DECEMBER 1985
USA $4.00
CANADA $4.50
A CWC/I PUBLICATION

80 micro

the magazine for TRS-80 users*

Reviewed in This Issue:

- Hyperzap
- Typitall
- MULTIDOS 80/64
- GBasic 3.0
- The Money Decision Series

THE ABCs OF C
Including YOUR OWN C INTERPRETER

TABLE TOPICS
How to Use Multiplan's Lookup Function

HOOP HOOPLA
The Ultimate Basketball Stats Program

WINDOWS IN BASIC!
The Hi-Res Board Makes It Simple

0 74470 65947 12

"Boy Am I Glad I Found You!"

People say this to us all the time. In fact, we'll go so far as to bet that if you spend a couple of minutes reading this article, you'll say the same thing.

Applied Creative Technology Inc. applies technology creatively. We produce machines that most computer dealers wouldn't *dare* tell you about... machines that deliver even *more* than what is expected of them (customers often tell us this too)... machines that can save you lots of money and headaches. Chances are if you do much computing at all, and use a printer or modem, you would benefit from having one of our products.

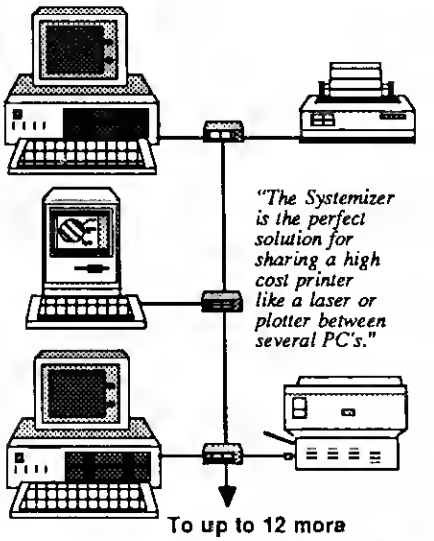
Enough of the promises... let's get to the facts.

"The Printer Optimizer has increased the performance of our system by 6000%, saving us thousands of dollars a year."

Our product line covers a myriad of applications. One product, the Printer Optimizer, is a *printer and modem control center*. It offers data spooling (using a 64K to 1 Meg buffer), the ability to connect several printers or modems to a single computer — without extra adaptors or software, and even the ability to modify or filter out data passing through it. A fellow from McDonnell Douglas told us: *"Every computer department ought to have a Printer Optimizer in their bag of tricks. It's great!"*.

The Printer Optimizer is also particularly useful to owners of laser printers. When you call for info, tell us if you have a laser printer — and if you do own a laser, you *should* call.

Another product, the Systemizer, allows several PC's or CRT's to share one or more printers.



Businesses ranging from small law firms to almost 100 of the *Fortune 500* are using Systemizers to save big dollars by eliminating printers and the office space and service costs associated with them. Now you can afford to own better printers like laser printers and plotters.

"The Systemizer is all the network many PC users need."

Jim Seymour, PC Week

The Systemizer is indeed the best solution for sharing printers you'll find. That's a bold statement, so we suggest you get a copy of our article *"The Wasted Buck Stops Here"* and see if you don't agree.

"Now that the Systemizer is available, buying a LAN to share printers is like buying an aircraft carrier to go water skiing!"

"Your 1 Megabyte printer buffer is a dream come true."

We also make a complete line of printer buffers and full fledged spoolers, with buffer capacities ranging from 64K to 1 Megabyte. Our latest creation is the **Buffer Box**. It's the lowest cost full-feature printer buffer available. Anybody who owns a printer should at least have one of these little wonders.

"Printing from a micro without a printer buffer is like trying to drain Hoover Dam with a soda straw."

You know, we hear the same refrain over and over: *"Geez! I wish I'd known about you before I bought..."*. In addition to the products mentioned, we also make a computer adapter for IBM Electronic Typewriters and some amazing boxes that adapt the Hewlett Packard LaserJet to various computers and word processing systems. Call us *now* before you waste any more time or money. You'll be glad you did.

Applied Creative Technology Inc.

2156 Northwest Hwy.
Dallas, TX 75220 USA

(800) 433-5373
(214) 556-2916
(TWX 9103332410 APPLCREATECH)

Did you know?
Only one spelling checker:

- works with both Scripsit and Superscripsit.
- integrates with all other popular word processing programs as well.
- requires *no special document placement!* Even runs on a single drive system.
- displays its dictionary so that you can find the correct spelling of words.
- offers integrated Hyphenation Option and Grammar & Style checker.

Electric Webster

SPELLING CHECKER ▪ AUTO-HYPHENATION ▪ GRAMMAR CHECKER

Displays Correct Spellings: If you don't know the correct spelling of a word, EW will look it up for you, and display the dictionary.

Verifies Corrections: If you think you know the correct spelling of a word, EW will check it for you before making the corrections.

Hyphenates Automatically: (Optional). Inserts discretionary hyphens throughout text.

Grammar & Style Checker: (Optional). Identifies 22 types of common errors. Makes suggested corrections with the stroke of a key. Runs within EW.

50,000 Word Dictionary: Uses only 2½ bytes per word; add as many words as you wish.

Fast Correcting: In as little as 30 seconds, Electric Webster can return you to your Word Processing program, with your text fully corrected and on your screen.

Integrates: with WordStar, Deskmate, Spellbinder, Volkswriter, Open Access, Allwrite, Newscript, Lazy Writer, Superscripsit, Scripsit, Electric Pencil, Copy Art, Powerscript, Zorlof, and LeScript (specify). Begins proofing at the stroke of a key; returns you to word processing automatically.

When ordering, stipulate word processing program and operating system.

"The Cadillac" of spelling checkers
— 80 Microcomputing, 9/82

ACCLAIMED:

"Electric Webster is the best. Just read any review in any magazine and I don't believe that you will find even one disagreement to that statement." *CINTUG, Cincinnati's Users Group Mag. 4/83*

"The most helpful program I've found is Electric Webster. After looking at nine proofreading programs, I've settled on Webster . . ." *Creative Computing 11/83*

This dictionary is not published by the original publishers of *Webster's Dictionary* or their successors

Performance "Excellent"; Documentation "Good"; Ease of Use "Excellent"; Error Handling "Excellent". *Info World, 8/82*

"Electric Webster, a fantastic spelling and grammar checker" *80 Micro 4/85*

Now see for yourself!

Try Before You Buy:

For the second straight year, 80 Micro readers have voted Electric Webster the #1 Spelling checker. Find out for yourself how accurate, fast and easy proof-reading can be. For only the $5 cost of postage, materials and handling, we will send you a special Electric Webster demonstration disk that works just like Webster, but proofs only half the alphabet. With it, you'll get a coupon worth $5 towards the purchase of Electric Webster.

LOW PRICES:

TRS-80, Model I/III or IV	$ 89.95
w/Correcting Feature	$149.95
Hyphenation	$ 49.95
Grammar & Style Checker	$ 49.95
CP/M, PC/DOS, Model 1000/2000 Electric Webster, w/Correcting Feature, Hyphenation, and Grammar	$169.95

Circle 45 on Reader Service card.

Cornucopia Software

POST OFFICE BOX 6111, ALBANY, CALIFORNIA 94706, PHONE (415) 524-8098

JMG

GREAT PROGRAMS, AND FREE SHIPPING TOO!

We've still got our very popular T/Maker and Word Processor deals continuing this month, as well as a new addition of T/Maker for the Tandy 1000 and other MSDOS machines. Since you'll probably be reading this as the holiday season begins to approach (at least for the retail business), I'm sure you'll want to consider us for that hard-to-please TRS-80 user on your gift list. And if you plan on giving an Infocom game, better get 'em soon because Infocom has discontinued production for the TRS-80 computers... it's a "while supplies last" basis from now on. I've run out of room now, so I'll leave you with this prediction: if Cadbury ever comes out with a computer, they'll probably use chocolate chips.

PASCAL-80

Easiest version of Pascal to learn! Editor and Compiler are already in memory. Nearly complete subset of standard PASCAL. Offers also many extensions to Pascal, including calls to machine language, screen control, random access files and more!

Models I/III (List $79) $59.50

PRONTO

Specifically designed for the 128K TRS-80 Model 4/4P Window controller program with many applications. Includes calculator, calendar, a sort utility, terminal facility, address cards, on-line help facility, and much more. "Sidekick" for the Model 4!

Models 4/4P $54.50

POWERDOT II

"The best high-resolution bit-image graphics program on the market for the TRS-80." Screen becomes a window on large full-page drawing board. AUTODRAW feature for lines and circles. POWERDOT II is 100% Machine Language. (Specify printer!)

MODELS I/III $29.50

PUBLIC DOMAIN DISKS

A fine collection of software from The Alternate Source!

Public Domain Disk (specify #1-#12) ..... Each $ 9.50
Public Domain Package #1-#6 ..... $49.50
Public Domain Package #7-#12 ..... $49.50
Public Domain Package #1-#12 ..... $89.50

EDAS/PRO-CREATE

One of Misosys' most popular utilities. Both a Full Screen Text Editor as well as a powerful Macro Assembler. Assembler supports nested macros, includes, and conditionals. Works excellently under most DOSes.

Models I/III and 4 ..... $69.50

SUPER UTILITY

"The indispensable first-aid kit for the TRS-80 users..." Contains over 60 different utilities for repairing, reviving dead files, reformatting, manipulation of files, and lots more!

Super Utility Plus (Models I & III) $59.50
Super Utility 4/4P & MSDOS $69.50

PACKAGE DEAL!

** MTERM **
** MSCRIPT **
** DOSPLUS IVa **

A complete operating system has just become very affordable! This new deal offers an operating system that is much faster and easier to use than TRSDOS. Not only is DOSPLUS IVa itself very user-friendly, it also offers a built-in menu driving system, and of course, GREATLY enhanced BASIC. Other included features of DOSPLUS IVa are: Text Editor, Linker, Assembler; Directory Verification/Repair, Disk Mapping, and File & Disk Editing. As if that is not enough, you now also get MSCRIPT with your purchase of DOSPLUS IVa. That's right, one of the easiest and most convenient to use word processors goes with your purchase. Also, MTERM Smart Terminal (one of the best full featured TRS-80 terminal programs available) is included in this deal. In addition to all of the remarkable features of MTERM, it will also enable you to log on to local Bulletin Boards and tell your friends about this fantastic deal!

DDSPLUS IVa / MSCRIPT / MTERM Package Deal
Models 4/4P (List $329.85) $159.50

SUPERCROSS XT

Designed specifically for transferring data and program files between TRS-80 disks and those of other computers

Models I/III or 4 ..... $94.50
with Convert Basic option $112.50

T/Maker™

• WORD PROCESSOR • SPREAD SHEET • GRAPHICS • • DATA BASE • & MORE!

This integrated software package for the Models 4/4P, as well as for MSDOS, combines many functions to become one of the best software deals available for any computer. Included are Word Processing, Spread Sheet Analysis (which provide a full range of mathematical functions), Relational Database Management (allows merging, multiple selection criteria, restructure of DataBase, Multiple Sorting etc.), Spelling Checker (55,000 word dictionary, correction feature, ability to create personal and professional dictionaries), Bar Chart Graphics (created directly from SpreadSheet data and supported on any printer), and finally, Data Encryption. If you are worried about learning T-Maker, worry no longer. It has excellent documentation and comes equipped with a Tutorial on the disk. Not only is it a great program, but it is also at a great price!!!

Model 4/4P (List $299) $194.50
MSDOS version (List $450) $294.50

LE SCRIPT

Great printer support, full Model 4 support and much more! On a 128K Model 4, you can have over 90K of text buffer for use on a single file. Model 4 features available while running in Model III mode. By far LeScript is our most popular program!

Models I/III or 4 (List $129.95) $94.50

WORD PROCESSING PACKAGE DEAL

LeScript and Electric Webster together!! Needless to say, these two great programs work excellently together!

W.P. Package (List $279.90) $199.50

ELECTRIC WEBSTER

Includes 50,000 word dictionary. Features fast checking, interactive correcting and personal dictionary expansion. (Specify computer and word processor when ordering)

Electric Webster (Models I/III or 4) (List $149.95) $119.50

Grammar or Hyphenation options (List $49.95) ..... Each $38.50

APPLICATIONS/BUSINESS

T-Maker (Model 4/4P)	$194.50
(MSDOS Ver.)	$294.50
TFC BBS	$ 99.50
Powerdot II	$ 29.50
POWERMAIL PLUS	$ 98.50
POWERMAIL PLUS w/Txt Merge	$128.50
LESCRIPT	$ 94.50
LESCRIPT CP/M	$149.50
LESCRIPT MS/DOS	$149.50
ZORLOF II	$ 49.50
MSCRIPT	$ 54.50
LAZYWRITER	$ 99.50
TYPEITALL	$ 99.50
PowerScript (New Version)	$ 34.50
PowerDriver	$ 24.50
Electric Pencil	$ 74.50
Electric Pencil w / Spell Check	$139.50
EOK Text Editor (Mod I/III)	$ 24.50
TEXTPRESS	$ 39.50
ELECTRIC WEBSTER	$119.50
E.W. Options (each)	$ 38.50
E.W. MS/DOS (Includes options)	$149.50
Datagraph with Pie Option	$109.50
Graphit	$ 34.50
PowerDraw	$ 34.50
Mumford's Disk Indexer	$ 34.50
Howe's System Diagnostic	$ 89.50
J & M's Disk Drive Analyzer - I	$ 84.50
J & M's Disk Driver Analyzer - III	$ 74.50
ENBASE Data Base Manager	$ 64.50
EDIT (Models I/III)	$ 18.50
Home Accountant (Model III)	$ 59.50
VersaLedger II (Models I/III)	$134.50
Versa Series (Models I/III) each	$ 89.50
TAS's ZMAIL Mail List	$ 24.50
PowerMail Plus	$ 98.50
Macro Typing Tutor	$ 24.50
Lazycomm Terminal	$ 29.50
Disk Term Terminal	$ 59.50
MTERM Smart Terminal	$ 59.50
DOSPLUS 3.5 (Models I/III)	$ 59.50
DOSPLUS IVA (Model 4/4P)	$119.50

GAMES

SUPREME RULER PLUS	$26.50
FLIGHT SIMULATOR	$29.50
NUCLIEX	$14.50
APE	$14.50
SIFTER SHIFTER	$ 8.50
BATTLE OF 2EIGHTY	$ 8.50
FOREST FIRE DISPATCH	$26.50
WARRIORS AND WARLOCKS (D&D Adv.)	$39.50
THE ADVENTURE SYSTEM	$34.50

THE BOOKSHELF

Using Super Utility	$14.50
ROM ROUTINES Documented	$16.50
Model III Assembly Language	$15.50
The C Programming Language	$17.50
Programmer's Guide to TRSDOS 6	$14.50
TRS-80 Disk and Other Mysteries	$16.50
Basic Decoded and Other Mysteries	$23.50
TRSDOS 2.3 Decoded	$23.50
Machine Language Disk I/O	$23.50
The Custom TRS-80	$23.50
How To Do It On the TRS-80	$23.50
Basic Faster and Better	$23.50
DFBLIB or BFBOEM Disks each	$17.50
Basic Disk I/O	$23.50
DFBLOAD Disk	$23.50

INFOCOM

Better be careful out there... Infocom's latest adventure seems to be the phasing out of their TRS-80 line. We will do our best to keep these popular games in stock, but once they run out, they are gone for good. Hitchhiker's Guide is our first casualty!

"Standard Level" ..... Each $34.50

PLANET FALL	WITNESS
ENCHANTER	CUTTHROATS

"Intermediate Level" ..... Each $39.50

ZORK II	ZORK III
SORCERER	INFIDEL

"Very Difficult Level" ..... Each $42.50

DEADLINE	STARCROSS
SUSPENDED	

UTILITIES

Alcor C Compiler	$ 84.50
Alcor Multi-Basic Compiler	$ 84.50
Super Utility 4/4P	$ 69.50
Super Utility 3.2	$ 59.50
Super Utility MSDOS	$ 69.50
Supercross XT	$ 94.50
Supercross XT w/CnvBasic	$112.50
Autoloader	$ 34.50
PRONTO (Model 4/4P)	$ 54.50
Other MISOSYS Utilities each	$ 23.50
Edas / PRO-CREATE	$ 69.50
DSMBLR III / PRO-DUCE	$ 23.50
Edas/Dsmblr Combo	$ 89.50
DIS' n' DATA I/III	$ 37.50
DIS' n' DATA (Model 4/4P)	$ 46.50
TASMON Monitor (Models I/III/4)	$ 34.50
Howe's Monitor #5	$ 19.50
CNVBASIC (Models I/III/4)	$ 27.50
Model 4 TOOLBELT	$ 39.50
TOOLBOX for LOOS	$ 39.50
TRAKCESS (Mod I)	$ 19.50
TRAKCESS (Mod I/II)	$ 24.50
PRO-ESP Utility Set (Model 4/4P)	$ 23.50
6.2 Plus Enhancements	$ 36.50
Impakt Utility	$ 29.50
NEWBASIC w/Analyst	$ 34.50
Analyst only	$ 19.50
ALE - Assembly Language Editor	$ 39.50
M-2AL Macro Assembler (Model III)	$ 79.50
Mumford's Instant Assembler	$ 44.50
Instant Assembler (Model 4/4P)	$ 59.50
2EN Assembler	$ 34.50
PASCAL 80 Compiler	$ 59.50
PASCAL 80 for CP/M	$ 35.50
LC / PRO-LC Compiler	$114.50
SBE Compiler	$ 46.50
ACCEL 3/4 Compiler	$ 44.50
2BASIC Compiler	$ 84.50
HartFORTH/PRO-HartFORTH	$ 69.50
Backrest Utility	$ 84.50
MULTIDOS Version 1.7	$ 79.50
MULTIDOS (Model 4/4P)	$ 89.50

ELECTRONIC NOTEBOOKS

KSOFT

SUPERLOG 4	$99.50
SUPERLOG 3 (I/III)	$99.50
LOG (Model I)	$44.50
LOG (Model III)	$44.50

MONTHLY SPOT LIGHT

ZBASIC 3.0

This long-awaited basic compiler is finally here! Enhancements included on this program include Device Independent Graphics, up to 54 digit numeric accuracy, a built-in interactive Editor and Compiler, structured Programming Constructs, and of course that is only scratching the surface. The nicest thing about ZBASIC is that the commands stay the same no matter what computer brand you use! Probably the best basic compiler around for any computer!

$84.50

CONVERSION PROGRAMS

BASIC 3 TO 4 CONVERT Model 4/4P ONLY (list $49.95)	$39.50
BASIC 4 TO 3 CONVERT Model I/III (list $49.95)	$39.50
BASIC GW CONVERT Model 4/4P ONLY (list $99.95)	$89.50
CONVERT BASIC Models I/III and 4 (list $29.95)	$27.50

OUR GUARANTEE:

We sell only top-quality software. If, however you are unsatisfied with a product, you may return it within 10 days (in good condition) for a refund, less $2.50 handling charge for programs under $50 ($5 for programs over $50). We also ask for a letter stating the reason for your return.

We will also beat any competitor's price by $1.00 (same conditions as competition, ie. shipping charges etc.) if you tell us where they advertise their price.

TO ORDER:

We accept orders by phone or mail. Specify your TRS-80 Model, exact program(s) wanted, and method of payment. We accept VISA, MasterCard, Check, and Money Orders (C.O.D. orders add $2.50 and Gov't Purchase Orders add $5.00). Electric Webster orders please specify Word Processor. Free shipping to continental U.S. and Canada. All prices are in U.S. Funds. Prices subject to change without notice.

ORDERS & INFORMATION

(416) 575-3201

10 a.m.- 7 p.m. Monday to Saturday

JMG SOFTWARE INTERNATIONAL

3235 Lockport Road
Niagara Falls,
N.Y. 14305

801 Mohawk Road West
Hamilton, Ontario
Canada L9C 6C2

(416) 575-3201

PUBLISHER
Peter Hutchinson

EDITOR-IN-CHIEF
Eric Maloney

MANAGING EDITOR
Peter E. McKie

SENIOR EDITOR
Penelope Hamblin

REVIEW EDITOR
Ryan Davis-Wright

COPY EDITORS
Marilyn G. McMaster
Trudy Nelson

TECHNICAL WRITERS
Bradford N. Dixon
Dave Rowell

TECHNICAL EDITORS
Marc-Anne Jarvela
Beverly Woodbury

LOAD 80 TECHNICAL EDITOR
Keith Johnson

EDITORIAL ADMINISTRATION
Carole Maclod
Kelly DeKoning

ASSOCIATE EDITORS
Hardin Brothers
David Engelhardt
John B. Harrell III
Terry Kepner
Thomas L. Quindry

ADVERTISING SALES

SALES MANAGER
William Smith

SALES REPRESENTATIVE
Michael Wozniak
1-800-441-4403

WEST COAST OFFICE
1060 Marsh Road
Menlo Park, CA 94025
415-328-3470

SALES REPRESENTATIVE
Alisson Walsh

ADVERTISING COORDINATOR
Judy Walker

ADVERTISING SECRETARY
Cathy Berry

MARKETING/PROMOTION DIRECTOR
Jane Butterfield

MPA ABC

The left bracket, [, replaces the up arrow used by Radio Shack to indicate exponentiation on our printouts. When entering programs published in *80 Micro*, you should make this change.

80 formats its program listings to run 64-characters wide, the way they look on your video screen. This accounts for the occasional wrap-around you will notice in our program listings. Don't let it throw you, particularly when entering assembly listings.

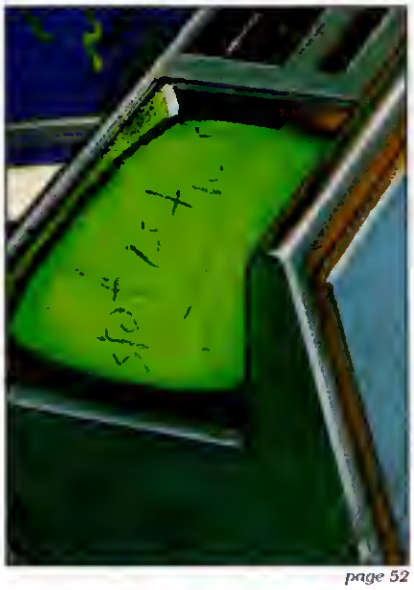
Article submissions from our readers are welcomed and encouraged. Inquires should be addressed to: Submissions Editor, 80 Pine Street, Peterborough, NH 03458. Include an SASE for a copy of "How to Write for *80 Micro*." Payment for accepted articles is made at a rate of approximately $50 per printed page; all rights are purchased.

*TRS-80, Scripsit, and TRSDOS are trademarks of Radio Shack, a division of Tandy Corp

80 Micro (ISSN-0744 7868) is published monthly by CW Communications/Peterborough Inc., 80 Pine St., Peterborough, NH, 03458. Phone: 603-924-9471. Second class postage paid at Peterborough, NH, and additional mailing offices. (Canadian second class mail registration number 9563.) Subscription rates in U.S. are $24.97 for one year, $38 for two years, and $53 for three years. In Canada and Mexico $27.97—one year only, U.S. funds drawn on a U.S. bank. Nationally distributed by International Circulation Distributors. Foreign subscriptions (surface mail), $44.97—one year only, U.S. funds drawn on a U.S. bank. Foreign subscriptions (air mail) please inquire. In South Africa contact *80 Micro* P.O. Box 782815, Sandton, South Africa 2146. All subscription correspondence should be addressed to *80 Micro*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737. Please include your address label with any correspondence. Postmaster: Send address changes to *80 Micro*, Subscription Services, P.O. Box 981, Farmingdale, NY 11737. Send Canadian changes of address to *80 Micro*, P.O. Box 1051, Fort Erie, Ontario L2A 5N8, Canada. Return postage guaranteed.

December 1985 80micro

page 38



page 52

page 58

On the Cover

38. **Born to Run** *by John B. Harrell III*
An introduction to C, the language that goes everywhere.

41. **Write Away** *by Daniel Zenzel Jr.*
All you need to run your own simple C programs. (Model 4; Load 80; Model 1000)

52. **Net Results** *by David H. Pleacher*
Our basketball statistics program shows you who's hot and who's not. (Models I, III, and 4; Load 80)

58. **Window Screens** *by Glen E. Sparks*
Painless hi-res Basic windows and pie charts. (Models III and 4; Load 80)

116. **On Displays: Sprucing Up Your Spreadsheet** *by John B. Harrell III*
Spreadsheet Beat investigates Multiplan's Lookup function and Lotus' colors.

Features

66. **Interrupt Anytime** *by Cary Oler*
Twelve programmable interrupts for TRSDOS 1.3. (Model III; Load 80)

74. **The Right Address** *by Maurice Dyke*
Follow these directions to get TRSDOS 6.X system addresses. (Model 4; Load 80)

76. **Rembrandt Redux** *by Dale Elton Rogerson*
Something extra for Model III users of our MacPaint-style graphics editor. (Model III; Load 80)

142. **1985 Articles Index**

142. **1985 Reviews Index**

144. **1985 Load 80 Index**

Departments

6. **Load 80 Directory**

8. **Side Tracks** *by Eric Maloney*

12. **Input**

14. **Feedback Loop** *by Terry Kepner*

21. **Pulse Train** *by Bradford N. Dixon*

25. **Reader Forum**

29. **Reviews**
GBasic 3.0, Draw, Joy-Mouse Interface
The Money Decisions Series
Hyperzap
MULTIDOS 80/64
Typitall

82. **Tidbit #29**

83. **Tidbit #30**

84. **Project 80** *by Roger C. Alford*

92. **Dave's MS-DOS Column** *by Dave Rowell*

102. **MS-DOS New Products**

106. **Basic Takes** *by Richard Ramella*

108. **The Next Step** *by Hardin Brothers*

116. **Spreadsheet Beat**

125. **Express Checkouts**
WordPerfect 4.0
Telecommuter
How to Use Your Radio Shack Printer
PRO-X-FTS

146. **New Products**

152. **Fine Lines**

LOAD 80

Load 80 gathers together selected programs from this issue of *80 Micro* and puts them on a magnetic medium for your convenience. It is available on tape or disk, and runs on the Models I, III, and 4.

Load 80 programs are ready to run, and can save you hours of time typing in and debugging listings. Load 80 also gives you access to Assembly-language programs if you don't have an editor/assembler. And, it helps you build a substantial software library.

Using Load 80 is simple. If you own a tape system, load the Load 80 tape as per the instructions provided. If you own a Model I or III disk system, you boot the Load 80 disk and transfer the files to a TRSDOS system disk according to simple on-screen directions. If you own a Model 4, copy the Model 4 programs from the Load 80 disk to your TRSDOS 6.X disk using the COPY command.

Not all programs will run on your system. Some Model III programs, for instance, will run on the Model 4 in the Model III mode, but not in the Model 4 mode. You should check the system requirements box that accompanies the article to find out what system configuration individual programs require.

If you have any questions about the programs, call Keith Johnson at 603-924-9471. Yearly subscriptions to Load 80 are $199.97 for disk, or $99.97 for cassette. Individual loaders are available on disk for $21.47 or on cassette for $11.47, including postage. To place a subscription order, or to ask questions about your subscription, please call us toll free at 1-800-343-0728 between 9 a.m. and 5 p.m. Or, you can write to Load 80, 80 Pine St., Peterborough, NH 03458.

Directory

C Trainer
Article: Write Away (p. 41).
System: Model 4, 64K RAM.
Basic C interpreter.
Language: Basic.
Cassette filespec: B.
Disk filespec: CTRAINER/BAS.

Hoops
Article: Net Results (p. 52).
System: Model III (Models I and 4 with changes), 32K RAM.
Basketball statistics program.
Language: Disk Basic.
Cassette filespec: C.
Disk filespec: HOOP/BAS.

Break In
Article: Interrupt Anytime (p. 66).
System: Model III, 48K RAM; Series 1 or Apparat editor/assembler.
Interrupts for TRSDOS 1.3.
Language: Assembly.
Cassette filespecs: BREAK (src), BREAKI (cmd), DEMO (cmd), SCROLL (src), SCROLL (cmd).
Disk filespecs: BREAKIN/SRC, BREAKIN/CMD, DEMO/CMD, SCROLL/SRC, SCROLL/CMD.

Locator
Article: The Right Address (p. 74).
System: Model 4/4P, 32K RAM.
Locate TRSDOS 6.X.X system addresses.
Language: Basic.
Cassette filespec: D.
Disk filespec: LOCATOR/BAS.

Windows
Article: Window Screens (p. 58).
System: Models III and 4, 48K RAM, high-resolution board.
Graphics and a pie chart application.
Language: BasicG.
Cassette filespecs: E, F, G, H.
Disk filespecs: SINEWAVE/BAS, PRISMRNG/BAS, VIEWPORT/BAS, WINDOWS/BAS.

Rembrandt
Article: Rembrandt Redux (p. 76).
System: Model III, 48K RAM; Series 1 or Apparat editor/assembler.
Screen dumps for graphics program.
Language: Assembly.
Cassette filespecs: LIST1 (src), LIST 2 (src).
Disk filespecs: LIST1/SRC, LIST2/SRC.

Page
Article: Tidbit #30 (p. 83).
System: Model 4, 64K RAM.
List files by line or screenful.
Language: Basic.
Cassette filespecs: I, PAGE (cmd).
Disk filespecs: PAGE/BAS, PAGE/CMD.

Convert
Article: Project 80 (p. 84).
System: Model 4 (Models I and III with changes), 32K RAM.
Converts object files to hex/ASCII.
Language: Basic
Cassette filespec: J.
Disk filespec: CONVERT/BAS.

Squeeze
Article: The Next Step (p. 108).
System: Model 4, 64K RAM.
Filter to condense debugged programs.
Cassette filespec: SQUEEZ (cmd).
Disk filespecs: SQUEEZE/SRC, SQUEEZE/FLT.

Delete
System: Models I and III, LDOS 5.1.
A multiple file kill command for LDOS 5.1.
Cassette filespec: DEL (cmd).
Disk filespec: DEL/CMD.

BAS = Basic   SRC = source code   CMD = object code

ART DIRECTOR
Beth Krommes
PRODUCTION SUPERVISOR
Dion Owens/Kanner
PRODUCTION ASSISTANT
Emily Hall
AD/GRAPHICS PRODUCTION
Gary Ciocci

GRAPHIC SERVICES MANAGER
Dennis Christensen
MANUFACTURING MANAGER
Susan Gross
FILM PREPARATION SUPERVISOR
Robert M. Villeneuve
TYPESETTING SUPERVISOR
Linda P. Canale

PRESIDENT/CEO
James S. Povec
VICE PRESIDENT OF PLANNING AND CIRCULATION
William P. Howard
VICE PRESIDENT/FINANCE
Roger Murphy
ASSISTANT GENERAL MANAGER
Matt Smith
ASSISTANT TO VP/FINANCE
Dominique Smith
CIRCULATION MANAGER
Frank Smith
DIRECT AND NEWSSTAND SALES MANAGER
Raino Wirein
1-800-343-0728
DIRECTOR OF CREDIT SALES
AND COLLECTION
William M. Boyer
EXECUTIVE CREATIVE DIRECTOR
Christine Destrempes

FOUNDER
Wayne Green

Cover photograph by White/Packert Photography

80 Micro is a member of the CW Communications/Inc. group, the world's largest publisher of computer-related information. The group publishes 57 computer publications in 20 major countries. Nine million people read one or more of the group's publications each month. Members of the group include: Argentina's *Computerworld/Argentina*; Asia's *The Asian Computerworld*; Australia's *Computerworld Australia*, *Australian PC World*, *Macworld* and *Directories*, Brazil's *DataNews* and *MicroMundo*; China's *China Computerworld*, Denmark's *Computerworld/Denmark*, *PC World* and *Run(Commodore)*; Finland's *Mikro*; France's *Le Monde Informatique*, *Golden* (Apple) and *OPC* (IBM); and *Distributique*; Germany's *Computerwoche*, *Microcomputerwelt*, *PC Welt*, *SoftwareMarkt*, *CW Edition/Seminar*, *Computer Business*, *Run* and *Apple's*, Italy's *Computerworld Italia*, and PC Magazine; Japan's *Computerworld Japan*; Mexico's *Computerworld/Mexico* and *CompuMundo*; The Netherlands' *Computerworld Benelux* and *PC World Benelux*, Norway's *Computerworld Norge*, *PC WORLD* and *Run (Commodore)*; Saudi Arabia's *Saudi Computerworld*; Spain's *Computerworld/Espana*, *Microsistemas/PC World*, *Commodore World*; Sweden's *ComputerSweden*, *Mikrodatorn*, and *Svenska PC*, the UK's *Computer Management*; *Computer News*, *PC Business World*, and *Computer Business Europe*, Venezuela's *Computerworld Venezuela*, the U.S.' *Computerworld*, *HOT CoCo*, *inCider*, *InfoWorld*, *MacWorld*, *Micro Marketworld*, *PC World*, *Run*, *73 Magazine*, *Focus Publications*, *On Communications*, and *80 Micro*.

Problems with Subscriptions: Send a description of the problem and your current and/or most recent address to: *80 Micro*, Subscription Department, P.O. Box 981, Farmingdale, NY 11737.
Problems with Load 80 Circulation. Address correspondence to Load 80, 80 Pine St., Peterborough, NH 03458.
Problems with Advertisers: Send a description of the problem and your current address to: *80 Micro*, Rt. 101 & Elm Street, Peterborough, NH 03458, ATTN.: Rita B. Rivard, Customer Service Manager. If urgent, call 1-800-441-4403.
Change of Address: Send old label or copy of old address and new address to: *80 Micro*, P.O. Box 981, Farmingdale, NY 11373. Please give eight weeks advance notice.
Microfilm: This publication is available in microform from University Microfilms International. United States address: 300 North Zeeb Road, Dept. P.R., Ann Arbor, MI 48106. Foreign address: 18 Bedford Row, Dept. P.R., London, WC1R4EJ, England.
Dealers: Contact Raino Wirein, Retail Sales Manager, *80 Micro*, Pine St., Peterborough, NH 03458 (800) 343-0728.

Radio Shack: Smurfin' USA

I'm spoiled. I'm so used to having dozens of computers at my disposal that I sometimes forget just how expensive they are. That's why I like to look through the Radio Shack sale fliers that occasionally come my way. They give me a new perspective on just how much a dollar really is these days.

Most recently, the Fall Sale flier landed on my desk, and it's loaded with bargains. My favorite section is the where-is-as-is sale. This is two pages stuffed full of discontinued gizmos and gadgets that Radio Shack is trying to unload—portable radios, telephones, walkie-talkies, and other electronic detritus.

The stars of this particular spread are the Model 100 and Model 4. The Model 100, placed at the top of the first left-hand page, is the main attraction—$299 for the 8K model. "Buy now for Christmas Giving!" exhorts the copy. The Model 4 is right below, at $299 for the 16K cassette version and $799 for 64K and two drives.

Now, $799 is a pretty good price for a full-blown computer. But as part of the where-is-as-is sale, the Model 4 begins to look pricey. For the cost of a 64K system, I could buy 40 Trim-Fones ($19.95 each), 114 cordless alarm clocks ($9.95 each), or 161 Smurf radios ($4.94 each). One hundred and sixty-one Smurf radios—now, there's something to think about. Having a computer in your home will scarcely get you a nod these days, but 161 blue Smurfs in your living room will make you the talk of the neighborhood.

The Model 4 and Model 100 aren't the only computer systems advertised in the flier. On the next-to-last page is the Tandy 1000 Personal Word Processing System—a Model 1000 with monitor, DMP-130 printer, DeskMate; and HomeworD word processor for $1,299. Overall, a pretty attractive deal.

But wait. This is even more expensive than the Model 4. For the extra $500, I could buy 101 more Smurf radios, enough to fill the kitchen, the bathroom, and part of my study.

OK, I don't really need 262 Smurf radios. But the where-is-as-is pages are gorged with other goodies. In fact, $1,299 will buy one of almost every item there. The list is practically endless:

Duofone talking home monitor	89.95
ET-280 pulse phone	24.95
ET-280 touch-tone phone	29.95
Duofone-332 phone dialer	29.95
Trim-Fone wall phone	19.95
TRC-84 CB walkie-talkie	14.95
Wallet/purse alarm	3.49
Door hanger alarm	4.95
PC-2 printer/cassette interface	49.95
PC-1 cassette interface	7.95
PC-1 printer/cassette interface	29.95
Wireless remote-controller	19.95
Wireless alarm system	59.95
Diskette storage box	14.95
CGP-115 color graphics printer	89.95
Cordless alarm clock	6.95
EC-2001 desktop calculator	14.95
16K Color Computer II	99.95
MC-10 16K RAM module	9.95
PortaVision AM/FM/TV radio	24.95
PocketVision LCD pocket TV	99.95
Realistic STA-204 receiver	129.95
Burger King radio	6.47
Smurf radio	4.94
Chronomatic clock radio	32.95
SCR-15 cassette recorder	79.95
Personal AM/FM stereo	21.95
Slim-Line pocket radio	6.95
CTR-51 desktop recorder	39.95
SCP-14 stereo cassette portable	34.95
Semiautomatic car antenna	22.95
Door-mount car speakers	19.95
Science-Fair digital computer kit	9.95
Science-Fair AM/FM radio kit	8.95
Video enhancer/stabilizer	59.95
AM/FM stereo portable radio	37.95
Super-Copter toy	12.95
3-D Sky Duel game	9.95
Total	1,288.15

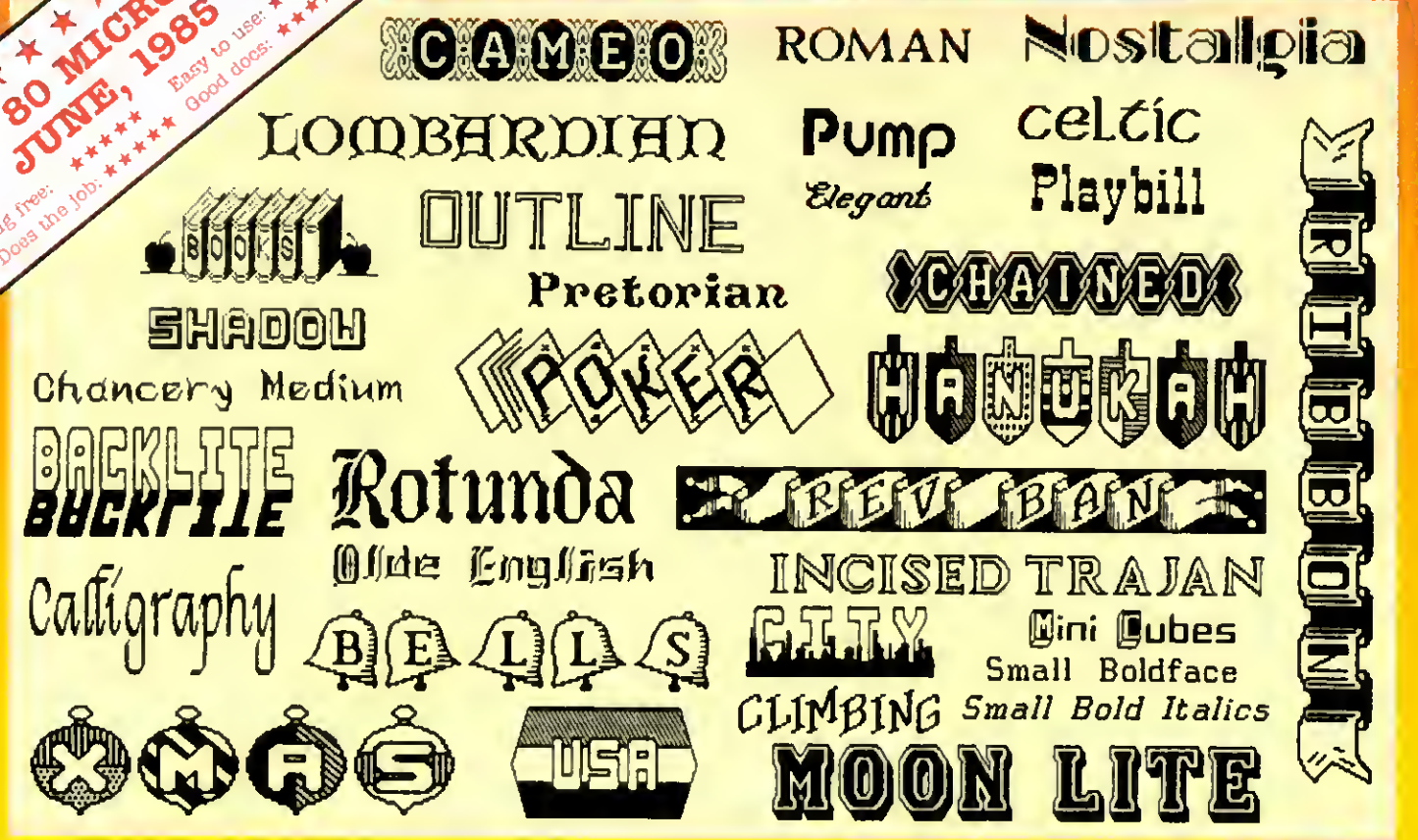
True, I don't need three phones, three cassette recorders, or eight radios. I'd be hard put to find much use for the MC-10 RAM module or Pocket Computer interfaces. But, what the heck, Christmas is coming up. Now's the time to give my mother that semiautomatic car antenna she's always wanted.

What the Radio Shack fliers ultimately prove is that we Americans can be perfectly happy with the simple pleasures of life, whether they be a $12.95 pair of fold-up headphones or a $6.88 Solar Energy Project Set that's a "great gift for kids 8 to 88." We owe a debt to Tandy for offering us such simple, low-cost alternatives to megabuck computing.

Trivia Time

One of CompuServe's TRS-80 special-interest groups recently produced an impressive thread devoted entirely to technical trivia of the TRS-80's early days. Here's a sampling: You'll find the answers on p. 82. If you get more than half, you can consider yourself a genuine old-timer.

1. Which system had only three error messages, and what were they?
2. You typed in SYSTEM and then followed the *? command with /12345 to do what?
3. What was the real update password on the TRSDOS 2.1 SYS files?
4. Finish this sentence: "Joe, you r----b------!" (Hint: this sentence was found in unlikely places on the first release of TRSDOS 1.3 disks.)
5. TRSDOS 2.2 and 2.3 included two programs called TEST1/CMD and TEST2/BAS. TEST1/CMD was a memory test program. TEST2/BAS was supposedly a "disk stress test program." In reality, TEST2/BAS was what Radio Shack program doctored up to look like it was actually doing something?
6. What did the initials of IJG, now-defunct publisher of the ". . .Other Mysteries" books, stand for?
7. Which DOS would not allow a Basic program to access a random file with a different LRL than that used to create the file?
8. Vern Hester wrote a DOS for the Model I that never became popular. What was it?
9. Level I Basic had only two string variables. They were fixed length. What were they and how many characters could they hold?
10. What was Level III Basic? ■



Circle 91 on Reader Service card.

DOTWRITER printed these on an Epson MX-80.

See What You Can Do With DOTWRITER!

DOTWRITER lets you create spectacular, eye-catching signs, invitations, letterheads, large sideways banners, catalogs, or even books. It is just what you need to turn your dot-matrix printer into a versatile typesetting machine. And it is available for your TRS-80 Model 4/4P (yes, in native mode), as well as for the Models I and III.

WHAT IS DOTWRITER?

DOTWRITER uses the "bit-image" graphics of your printer to produce the kinds of stunning results shown inside the box. It is a full-function text printing program, so you can inter-mix different character sets, do centering, paragraphs, pagination, magnification, draw horizontal and vertical lines, reversals (black on white), and even print right-justified proportional text.

DOTWRITER includes the printing program, complete documentation, and fourteen useful typefaces (60 to 90 characters per typeface). We will include the 170-page Letterset Reference summary at half-price ($10.00) with your order.

To use DOTWRITER, just write your text with any popular TRS-80 Word Processor (such as ALLWRITE or SuperScripsit), add the necessary formatting commands, and DOTWRITER will do the rest.

36 more letterset disks are available separately. Each has 3–12 complete typefaces. The disks costs less than $25 each and you may purchase them at any time.

SIDEWAYS SPREADSHEETS

If your VisiCalc spreadsheets are too wide for your printer, our "LONGVIEW" option may be just what you need. It is an add-on that turns spreadsheets sideways so that DOTWRITER can print them down the page instead of across. LONGVIEW comes with three additional fonts.

EQUIPMENT REQUIREMENTS

DOTWRITER needs a TRS-80 I, III, 4 or 4P with 2 disk drives and 48K of memory. Separate versions of DOTWRITER support EPSON MX-80 with Graftrax, MX-100 with Graftrax-Plus, and FX, JX, RX; C.ITOH 8510/1550; MICROLINE 84-2/92/93; RADIO SHACK DMP 110-2100/CGP-220; GEMINI 10X/15X and other STAR printers.

We printed our samples on an Epson; sizes may vary on other printers. Many of the fonts shown above are available at extra cost.

Send for free print samples! We've only shown you a few of the 240 DOTWRITER fonts. For the best in TRS-80 graphics printing, we suggest you order DOTWRITER today, toll-free.

Please specify Printer and Computer when ordering.

DOTWRITER	$99.95
LONGVIEW	29.95
Additional Letterset disks (3–12 fonts/disk)	24.95
	3 for 49.95
Letterset Reference Book	20.00

FREE **bonus disk with two Banner fonts when you order DOTWRITER!**

ORDER NOW, TOLL-FREE
(800) 824-7888, oper. 422

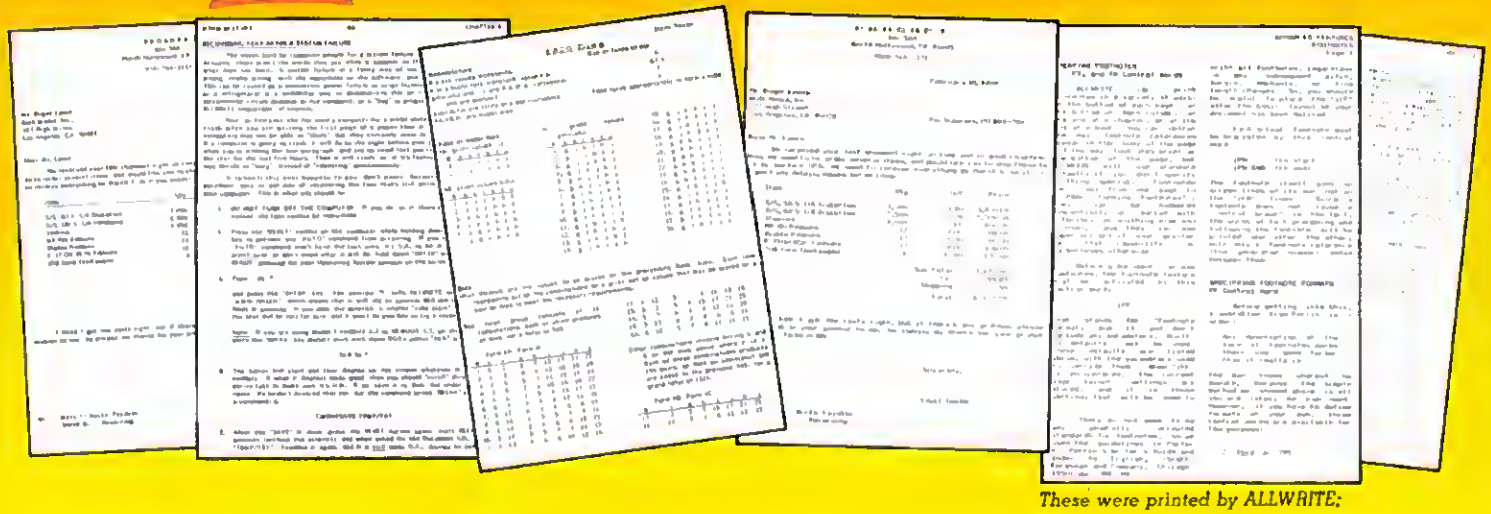
PROSOFT®

Dept. C, Box 560, No. Hollywood, CA 91603
(818) 764-3131 Information and Same-Day Processing

TERMS: VISA, MC, checks, COD. Please add $3.00 shipping in U.S. or Canada. Sales tax in CA. Most orders filled within one day.

Allwrite!™

The Premier Word Processor for Your TRS-80 Model I, III, or 4



These were printed by ALLWRITE; shown 20% actual size.

We are proud to offer you the one Word Processor that will satisfy all your writing needs: ALLWRITE. It sets new standards for text editing and printing, and will give new life to your TRS-80. Let us tell you why . . .

In an attempt to push the public into expensive 16-bit computers, many manufacturers have been saying that the TRS-80 is obsolete. The truth is that the software, not the hardware, makes the difference. And the best word processor of all is now available *only* on the humble TRS-80, not on those expensive 16-bit machines!

ALLWRITE will save you time and let you produce the highest-quality, most professional-looking letters, term papers, and reports available on a micro-computer.

Allwrite Can Save You Time!

Reads a 25,000 character file (10 printed pages) from disk in SIX SECONDS . . . does a global search-and-replace in FOUR SECONDS . . . outruns even the fastest popular micro-printer.

ALLWRITE'S Screen Handling Makes Word Processing Easier Than Ever

Change text width at any time; wide lines shift left and right as you type. ALLWRITE preserves double-blanks between sentences, uses the entire screen for text, and displays a complete Status Screen at the touch of a key. Scroll by line, partial screen, full screen, to top or end of file, or to any marked point. Move cursor by character, word, tab, line, or screen.

You can set and change on-screen tabs and store them on disk. The print-time tabbing features are incredibly versatile: they allow left, right, and centered tabs, and even line up your decimal points.

ALLWRITE shows you where you forgot to turn off underlining, boldface, italics, or double-width. Special on-screen Preview feature shows page breaks and page layouts . . . including underlining and boldface. In "Summary" mode, ALLWRITE quickly flags formatting errors without wasting time printing all the text. These standard features make document preparation faster and easier than ever!

State-Of-The-Art File Handling

There is no upper-limit on document size with ALLWRITE, because it chains files *backwards* as well as forwards, even across diskettes. Switch from one chained file to another in less than six seconds by pressing two keys. Select portions of other files for inclusion at print time . . . great for stock paragraphs.

ALLWRITE salvages text from bad disks! If a sector goes bad, you won't lose the entire file, because it

TAKES FULL ADVANTAGE OF YOUR MODEL 4.

The model 4 version of ALLWRITE uses **the entire 80-by-24 screen.** On a 64K machine, you can edit **over 34,000 characters** of text. On a 128K machine, you can edit **THREE FILES AT THE SAME TIME!** The second and third files can be over 32,600 characters each, for a total of **almost 100,000 characters** of text in memory.

will skip bad sectors, read the rest of the file, and then show you where the lost text belongs. This advanced error recovery turns a disaster into a feeling of profound relief.

User-Definable Soft Keys Reduce Typing Time

You can store 22 phrases or commands at a time into "soft-keys," then press just two keys to retrieve them. This makes frequently-used phrases and formatting controls a snap to use. You can store these definitions on disk and build a library of hundreds of pre-programmed keys to fit every one of your applications.

Our specially-designed templates fit right on your keyboard to let you see your settings at all times. Each template is also a Reference ("Cue") Card, so it is always right in front of you when you need it, without using up valuable screen space.

ALLWRITE Is Easy To Learn

ALLWRITE's commands and control keys are easy to remember because they use the first letters of common English words: 'CE' stands for 'Center,' 'Search' and 'Replace' do just that, and so forth. The on-line HELP menu offers over fifty screens of topics.

ALLWRITE's superb documentation will get you started quickly. Portions of it are designed for beginners, with every feature clearly explained in step-by-step tutorial style. Since you won't always be a beginner, other parts of the book offer advanced topics. There is a cross-reference summary chapter, a 14-page comprehensive index, and a detailed Table of Contents. We've been developing computer programs and manuals for over 23 years, and understand the importance of good documentation.

ALLWRITE works with all major DOS's on Models I, 3, and 4/4P.

PROSOFT'S On-Going Customer Support

Perhaps the best reason of all for having ALLWRITE is the continuing support we offer you: friendly, expert, direct support that is unsurpassed in the micro-computer industry.

Note to students: with its Footnote, Table of Contents and Index features, ALLWRITE is ideal for your reports and Term papers.

Note to teachers: ALLWRITE makes it very easy to generate multiple-choice exams and answer keys. Ask for free instructions when ordering.

"ALLWRITE is a professional system that sets a new standard in word processing. It's powerful and easy to learn and use."
80 MICRO, Nov., 1984

Customer Comments

"This is the best software package I have ever received . . . superb, easy to use, fast, and has more features than the business word-processor at the office." *(E.R.L.)*

"Your company and products have to be one of the strongest factors I can think of for keeping me with the TRS-80!" *(J.R.H.)*

"NEWSCRIPT is the Cadillac of word processors. ALLWRITE is the Mercedes Benz!!" *(B.E.)*

". . . a very readable manual." *(D.S.)*

BENEFITS OF OWNING ★ ★ ALLWRITE ★ ★

If Word Processing is important to you, PROSOFT's ALLWRITE is the best choice you can make. The clean, professional appearance it adds to your letters and reports will make an excellent impression on people. We will be happy to send you free print samples so that you can see for yourself how good ALLWRITE will make you look.

You probably know that quality word processors for CP/M and the IBM-PC sell for $300–500, and they don't have ALLWRITE's capabilities or speed . . . or PROSOFT's proven, on-going support. Now, for a fraction of the cost of a new computer, you can have the most complete word processor of all. And you won't have the headaches of starting all over again with a new, different computer.

HUNDREDS OF USEFUL CAPABILITIES

ALLWRITE comes with just about every useful word processing feature . . . standard. Here are some highlights: excellent right-justified proportional printing on most printers having that ability; powerful Form Letter and Mailing Label preparation; Instant counts of words, characters, lines, changes; block Move, Copy, Delete, Putfile, Getfile, and List; delete by character, word, line, sentence, paragraph, or block; insert and one-key insert; great RS-232 printer support; accepts all 256 ASCII codes from keyboard; intermix pitches on same line (printer-dependent); 1.5 line spacing, 6, 7, 8, 12 lines per inch (printer-dependent); does multiple-columns on all printers; perfect alignment of hanging indents; variables, logic statements, conditional printing; wildcard Directories; integrated with Electric Webster and DOTWRITER for Models I, III, and 4 (these are sold separately); "Legal" line numbering; paragraph, list, and figure numbering; supports most popular printers (all "printer drivers" included); compatible with high-memory drivers; fully explains all DOS and ALLWRITE error messages; wildcard search-replace; tabs, search-replace, other settings remembered across files; word reversal; up to nine levels of boldface; flexible page titles; footnotes at bottom of page or end of document; Table of Contents and Index generation; and PROSOFT's unmatched text formatting and printing capabilities.

How To Order

You can order by phone or mail. For quickest delivery, call our Technical Support line. Please specify your TRS-80 model (I, III, or 4, 48K, at least two disk drives), and your printer(s). Our price includes normal shipping in the U.S. and Canada. The sooner you order, the sooner you will begin to benefit from the ALLWRITE! Word Processor.

Allwrite for the TRS-80

$199.95

Circle 30 on Reader Service card.

ORDER NOW, TOLL-FREE

(800) 824-7888, oper. 422

PROSOFT.®

Box 560, No. Hollywood, CA 91603

(818) 764-3131 for Technical Information and Same-Day Processing.

Terms: VISA, MasterCard, checks, C.O.D. California residents please add sales tax. Most orders filled within one day.

Writer's Block Letters

I agree with Eric Maloney (Side Tracks, August 1985, p. 8): Word processors do not improve writing. I always used to write quickly and easily with a pen or typewriter, but Scripsit brought about writer's block. All those editing commands intimidated me while I was writing a first draft; I spent too much time tinkering and too little time writing.

Now I use Delmer D. Hinrichs' Basic Word Processor ("The Return of Hinrichs' Word Processor," March 1984, p. 100) almost exclusively. It has many editing commands, but they're off in the edit mode, where they should be. While I'm writing, I have one editing command at my disposal—the backspace. And that, too, is how it should be.

80 Micro has published several different versions of Hinrichs' program. The one I use appeared in the January 1983 issue (p. 200). If you use it in conjunction with Mark Goodwin's type-ahead utility ("Getting Ahead," July 1985, p. 65), you'll find keyboard response excellent. And since Hinrichs wrote his program in Basic, you can easily modify it to suit your needs.

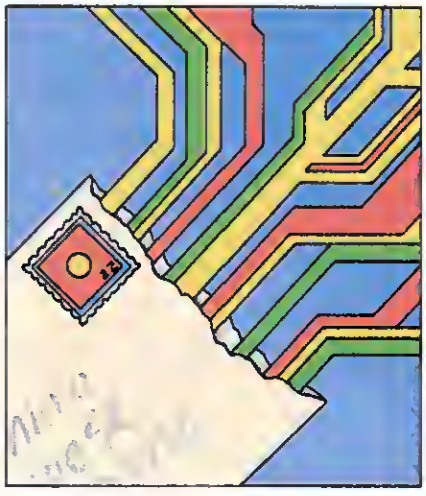
Mark Allen Reed
West Lebanon, NH

I'm delighted that Eric Maloney plans to return to his manual typewriter since he seems to have a need to indulge himself in some kind of language orgy. Also, since he says he has to struggle to express himself with a word processor, he should stop using one. However, what he believes is true for him clearly is not true for most of the rest of us. That word processors don't improve writing for most people is absurd.

The notion that word processors can liberate creative writing by removing most of the restrictions imposed by the pencil or typewriter is completely accurate in the view of many people who are much more productive after using a word processor.

Anyone experienced with word processors knows that they do not supply creativity to writers who have none, but they do make writing creatively much easier and more efficient than any other method known.

Murlon H. Dye
Commerce, TX



Eric Maloney's observation about word processors is quite accurate. One should also note that a word processor will not directly improve a writer's style if he lacks it. Word processors do make rewriting and editing a breeze, especially with large amounts of copy, but they're no substitute for a command of the language, punctuation, ability, and innovation.

I disagree, however, that they can injure writing skills. The person pushing the pencil or tapping the keys will determine the worth of the creation. If Maloney finds he's more creative with an ordinary typewriter than with a word processor, perhaps he's suffering from a case of cursor-blinksis-anxiety, a recently discovered emotional disorder brought about by the eternal, unrelenting blink of screen cursor that reaches into a person's subconscious with the hidden message, "Come on! Come on! What's the next word! sentence! paragraph!"

Jim Merlini
Montgomery, AL

My cursor likes to hum old Smokey Robinson tunes.

—E.M.

80 Micro's BBS is open 24 hours a day. It offers programs you can up- and download, special-interest groups, and a classified section. You can reach the board at 603-924-6985; UART settings are 300/1200 baud, 8-bit words, 1 stop bit, no parity.

Basic Solution

In your August 1985 issue you ran articles on Model 4 Basic (p. 38) and GW-Basic (p. 46). Both articles touted the use of the Common and Chain statements to link Basic programs. In practice, I've found both statements useless. The problem is that you have to save the programs you want to chain in ASCII format. If a program is so long that you have to separate it into smaller programs, the individual modules take so long to load as to be impractical.

I think it's faster to save the programs in compressed form and ssve to a disk file the variables you want to pass. Then the succeeding program, linked to the first one by a Run statement, can reload the variables.

For even greater speed, you can save the variables to a RAM disk. The variable-passing routines found in Lewis Rosenfelder's *Basic Faster and Better* work well in Model III mode on a Model 4, but they won't work with Model 4 Basic or GW-Basic.

William D. Tabor Jr.
Thibodaux, LA

Window-Comments

Thank you for the favorable review of our product, Window-Comm (November 1985, p. 31). One thing the review didn't mention was that Pacific Software Consultants offers a $10 rebate to each customer who persuades a friend to buy the product (limit one per purchased copy), making Window-Comm an exceptional value at $8.95 after the rebate. A friend sold on it need only include the software license number of the original purchaser along with his order and we'll send the latter a $10 rebate check.

The other thing you should know is that, while Window-Comm had been running on the Model III, we released a Model 4 version in October 1985. It offers several enhancements. All Model 4 owners who bought the Model III version will receive the Model 4 version free of charge.

Stephen W. Apple
Pacific Software Consultants
San Luis Rey, CA

Send your correspondence to Input, 80 Micro, 80 Pine St., Peterborough, NH 03458.

Illustration by Mark Fisher

CHRISTMAS SPECIALS

The ALPHA SPEECH SYNTHESIZER

Outstanding performance and value for only:

This is your chance to experience the power and pleasure that speech adds to your TRS-80. If you could read the thousands of testimonials we have received you would be convinced. Instead, our unconditional 15-day money back guarantee fully protects you.
Watch your friends faces when your TRS-80 starts talking.

$49.95*

Thousands sold at $75.90
*When purchased with text to speech software.

TALKER 4.0

Unlimited vocabulary Text-to-Speech Software. Powerful, yet easy to use; even non-programmers can enjoy it. Talker 4.0 features:

- Automatic video and/or keyboard echo (if you want it).
- Pitch control • Voice-speed control • Spelling mode
- Says numbers (up to 999 trillion) • Simple BASIC commands • Much more!

Only $39.95

Small Print: **Hardware** Power supply, speaker and manual included. Model I unit plugs into keyboard or expansion interface 40 pin bus. Model III,4,4P unit plugs into 50 pin I/O bus. Model 4P needs short 50 pin extension cable $14.95. Use our "Y cable" (see next page) if your bus is already used. **Software** Works with all DOSes (not CPM), is 6.2K long, and relocates itself to the top of available memory. **Manual** available for $5.

Dr. SIGMUND

Artificial Intelligence at work! If you want to show off your computer, run "Dr. SIGMUND" and see their expressions as your TRS-80 has an intelligent conversation with you. Even you will be impressed!

PERSONALITY TEST

By Dr. James E. Hord, Jr. for your ultimate entertainment. This elaborate personality test will amaze you, and puzzle your friends. Besides talking to you, it will print a painfully accurate report.

TALKING WORD PROCESSOR

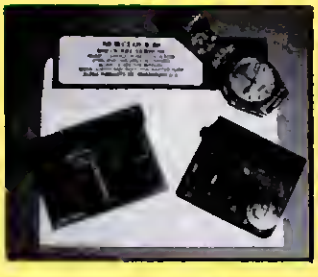
By George McCoy of Rehab Research. The Alpha Speech Synthesizer was chosen for this functional word processor with full speech capability. A perfect example of computer speech.

Each of these three programs require 48K and are available on disk only. The Alpha Speech synthesizer is required for speech. Each program is only $29.95

SPECIAL: ALL THREE FOR ONLY $59.95

NEWCLOCK

Model I $39.95
Model III,4 $59.95



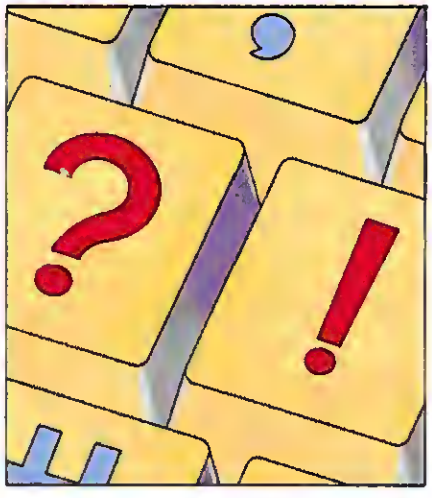
The right time at the right price! Keep the time and date with quartz accuracy, even when your computer is off. The backup lithium battery (included) will last for over 2 years. Software on tape or disk, please specify. Use "TIMESET" once to set the clock. Use "SETCLK" to set your computer's internal clock (at power up) or use "TSTRING" so that the "TIME$" function reads the Newclock.
Connection: Model I: plugs into the keyboard or expansion interface. Model III,4,4P: plugs into the 50-pin I/O bus. 4P needs short 50-pin extension cable $14.95 Compatible with all operating systems.

ALPHA Products
79-04 Jamaica Ave., Woodhaven, NY 11421

800-221-0916
Orders Only. NY& info call (718) 296-5916
Hours: 9-5 Eastern Time
Circle 17 on Reader Service card.

Add $3.00 per order for shipping. We accept Visa, MC, checks, M.O. C.O.D. add $3.00 extra. N.Y. residents add sales tax. Shipping to Canada is $5.00 Overseas, FPO, APO add 10%

Send your questions or problems dealing with any area of Tandy/Radio Shack microcomputing to Feedback Loop, 80 Micro, 80 Pine St., Peterborough, NH 03458.



Q: Thank you for including the kind remarks concerning TBase, my subroutines for recording data on cassette-based TRS-80 computers (July 1985, p. 17). Unfortunately, I have received a letter from Bruce O'Connor, a lawyer in Seattle, WA, who complains that my program name is too similar to a trademark of his client's, Traveling Software Inc. I have thus renamed my product Tapestry, and am including a copy of O'Connor's letter in the manual distributed with each copy of my work. *(David B. Dillon, Derwood, MD)*

A: For those who missed Dillon's letter in the July 1985 Feedback Loop, he has developed a set of 19 Assembly-language routines that let Model III Basic maintain a cassette-based data file much the same way that Disk Basic maintains a random-access ASCII file. For more information, contact Dillon at 16533 Baederwood Lane, Derwood, MD 20855.

Q: In the July 1985 Feedback Loop (p. 16), Ralph Turner asked for help in using cassette Scripsit 3.1 with his DMP-200 printer. I think I have a patch he could use. It isn't particularly elegant, but it gets the job done with a minimum of trouble.

My patch occupies Scripsit's title area, so that it steals no memory from your text. I used a method suggested by Arne Rohde's VCMOD utility (April 1983, p. 210). It lets you send control codes to your printer by intercepting every less-than sign it encounters and Anding the ASCII value of the following character with 31. (In other words, the program keeps subtracting 32 from the ASCII value until the result is itself below 32.) Using this method, you can send the escape character (CHR$(27)) to the printer by embedding <; or <(into the text; the BEL character (CHR$(7)) by embedding <', <G, or <*q*; and so on.

In addition, if you want to print characters with ASCII values above 127, simply embed a greater-than sign in the text, followed by the character with an ASCII value of 128 less than that of the character you want to print. To print CHR$(240), embed >"; to print CHR$(191), embed >?; and so on.

Keep in mind that the less-than and greater-than signs foul up Scripsit's justification routines. I'd suggest setting J = N at the beginning of your document. And don't be frightened by the mention of Anding ASCII values—with your printer's ASCII code charts nearby, and a half-hour or so of practice, you'll soon get the hang of it.

Use a high-memory monitor (or EDTASM) to enter this program into memory after you load Scripsit, then transfer control to Scripsit's entry address, 4303 hexadecimal (hex). In addition, change memory locations EEF hex and 4EF7 hex from CD 3B 00 to CD BD 48. Be sure to change these addresses before transferring control to Scripsit. This final alteration installs the patch. *(Mark Reed, West Lebanon, NH)*

A: Thank you for sending in your Scripsit patch (see the Program Listing).

Q: I have a Level II 16K Model I, and I have just upgraded to a Model 4P. I want to transfer all my old programs to my new computer. Here in Chile some special chips (Signetics 2681) are not available and it is not easy to build an RS-232 interface for my Model I. I found Bob Hart's article "Bare Bones Communicator" in the June/July 1982 issue of *80 Micro* (p. 128). I built the circuit and it worked. . .in one way. I can transfer Basic programs from the Model I to Model 4P but the Model I doesn't acknowledge Model 4P signals. I suspect the XRX modification in my old machine is the problem. I have read about that mod, but I don't know what

Program Listing. A Scripsit patch program.

```
; Rand Assembly of "Patch"
;     written by Mark Allen Reed for Model III cassette
;      Scripsit, version 3.1
;     assembled to begin at location 4BBD8, Scripsit's
;      title area
;
PATCH   PUSH    AF              F5
        LD      A, (FLAG)       3A E7 48
        OR      A               B7
        JR      NZ,NEWPRT       20 11
        POP     AF              F1
        CP      '<'             FE 3C
        JR      Z,SAVFLG        20 08
        CP      '>'             FE 3E
        JR      Z,SAVFLG        28 04
LOOP    CALL    0033BH          CD 3B 00
        XOR     A               AF
SAVFLG  LD      (FLAG),A        32 E7 48
        RET                     C9
NEWPRT  CP      '<'             FE 3C
        JR      NZ,GRAPHC       20 05
        POP     AF              F1
        AND     1FH             E6 1F
        JR      LOOP            18 EF
GRAPHC  CP      '>'             FE 3E
        JR      NZ,LOOP         20 EB
        POP     AF              F1
        OR      80H             F6 00
        JR      LOOP            16 E6
FLAG    DEFB    8               08
```

End

Illustration by Mark Fisher

The Amazing A-BUS

NEW

Hobbyists, Engineers, Scientists, OEMs, universities, the A-BUS is for you!

What is the A-BUS? The A-BUS is the best way to connect a variety of **Input** and **Output** cards (such as analog converters, relays, sensors, motor controllers, etc.) to your computer.

A typical **A-BUS** system consists of: • An adapter card and cable to connect your computer to the **A-BUS** standard • The **A-BUS** motherboard, with several slots in which you plug the different Input and Output cards. • Your choice of cards listed below, depending on your application. (Many more cards will be released soon.)

The "A" stands for Amazing, and here is why:

❶ The **A-BUS** works with any TRS-80 models I, III, 4, 4P, 4D, 1000, even 100, 200 and CoCo. In addition, it will also work with IBM or Apple computers. Should you ever move to another system, your investment is protected. Only the low cost adapter card has to be changed!

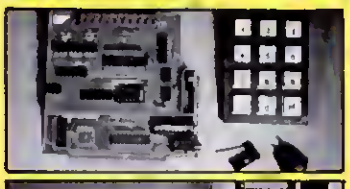
❷ The system is expandable to meet current and future needs easily.

❸ Low cost and reliability will ensure your project success.

A-BUS Adapter for Model I Plugs into 40-pin I/O card edge (on KB or E/I)	AR-131...$39
A-BUS Adapter for Models 3,4,4P,4D Plugs into 50-pin I/O bus.	AR-132...$49
Cable (3 ft.) Computer to **A-BUS**	CA-163...$29
A-BUS Motherboard, for up to 5 cards (not needed it using only one card)	MB-120...$99

A-BUS new Relay Card: RE-140...$129

This industrial grade output card includes 8 relays. (Contact rated 2 Amp @ 125V) All the decoding necessary is included which means that you can connect up to 64 cards (which is 512 relays.) Easily controlled using "OUT" commands. For example OUT 0,0 turns all the relays off on card #0. Eight LED's show the states of the relays.



A-BUS new Isolated Input Card: IN-141...$49

This optically isolated input card makes it safe and easy to connect external devices (switches, sensors, thermostats, keypads) to your computer. Simple INP commands read the status of the eight inputs. Full address decoding allows up to 64 input cards (that's 512 channels) per computer.

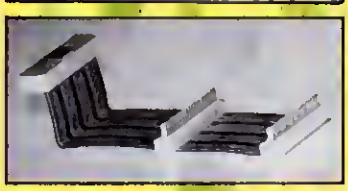
A-BUS new Analog Input card: AD-142...$119

8 channel 8 bit Analog to Digital converter. Your computer can read voltages, temperatures, pressures, light levels, etc. • Input range: 0 to 5.1 Volts. • Resolution: 20mV. • Conversion time: 120 microseconds. In BASIC, you can take up to 100 readings per second. • Port address: selectable. Up to 64 Analog-80's can be connected to your computer for a total of 512 channels!

A-BUS new Dual Stepper Controller: ST-143...$69

Don't be afraid of stepper motors anymore. The special package (below) includes everything you need to get familiar with steppers: • Controller card drives 2 steppers (12V bidirectional) ST-143...$69 • Stepper: 48 steps per revolution, up to 300 steps/second. MO-103...$15 • Power supply PS-126...$10

Special Package: Controller, *two* steppers and power supply: PA-181 .....$99



Special Cables

Disk drive extender cable (8")...C160:$9.95

Y-Cable for Mod I bus (40 pin): • **X2-40**...$29 • **X3-40**...$44 • **X4**...$59 • **X5**..$74

Y-Cable for Mod 3 & 4 bus (50-pin): • **X2-50**...$34 • **X3-50**...$49 • **X4-50**...$64

Disk drive cable (34 pin): • 2-drive...**C162:**$32 • 4-drive...**C163:**$45

Our cables are made with high quality gold plated connectors to ensure utmost reliability.

Green Screen.....$12.50

Do your eyes a favor, put on a green screen. Tens of thousands are in use because they work. Contrast is enhanced and eye fatigue is greatly reduced. Our green screen is curved; it fits right on the face of the tube. (Fits Models I,II,III,4,12,16)

Printer-Switcher.....$59

A must if you have two printers, plotters, or any devices using the standard parallel printer port. End the hassle of plugging and unplugging cables. You can select either device at the flick of a switch. For Models I,III,4,4P,4D.

ALPHA Products

79-04 Jamaica Ave., Woodhaven, NY 11421

800-221-0916

Orders Only. NY& info call (718) 296-5916

Hours: 9-5 Eastern Time

Circle 17 on Reader Service card.

Add $3.00 per order for shipping. We accept Visa, MC, checks, M.O. C.O.D. add $3.00 extra. N.Y. residents add sales tax. Shipping to Canada is $5.00. Overseas, FPO, APO add 10%

it is. My model I has the serial number 058836 and two NEC ROM chips (8043364 and 8043732): the initial prompt is "Memory size?" Can it have the XRX modification? If it has, how can I disable it momentarily? *(Jorge Herrero Endesa, Los Angeles, Chile)*

A: Before you start looking at the XRX modification, alter the DB-25 connector you're using by tying lines 6, 8, and 20 together. The problem may be that the Model 4P is waiting for your Model I to transmit a Clear-to-send or Data Terminal Ready signal. Because the bare-bones communicator doesn't have those lines attached to anything, the Model 4P thinks the Model I isn't ready to receive, so it sends nothing.

Next, the XRX-III modification improves the reliability of the cassette file-loading procedure by making allowances for the low-quality cassette units and tapes on the market. It does, however, mean you can't use the cassette port at speeds other than 500 baud. If you have the XRX modification installed, and tying lines 6, 8, and 20 together doesn't help your RS-232 communications, then you must disable the XRX-III modification.

Information about the XRX-III modification and the standard TRS-80 cassette circuitry is available in Dennis Kitsz's book *The Custom TRS-80 & Other Mysteries*. This book is currently available from Montezuma Micro, an *80 Micro* advertiser.

Q: I bought a Model 4P and haven't been able to find programs for it in *80 Micro*. Can you explain why? *(Kenneth Fonseca, Los Angeles, CA)*

A: One thing to keep in mind about the Model 4P is that all Model III and many Model I programs will run on it. While the program listings might not explicitly say 4P, most will work fine. Also, a great many of the more powerful programs for the Model 4 series are written in machine-language, such as Hardin Brothers' windowing program for the Model 4 (June, July, and August 1985, p. 102, 100, and 98, respectively).

Q: The letter from Jon C. Schultz in your August column (p. 16) concerned a Radio Shack disk drive he bought in Japan that he can't get repaired. Your answer was to write to Tandy's Japanese division and ask for a service manual. I believe I can help him. I don't have the service manual for the specific drive he has but I'm quite sure that I can repair it for $50 or less if he will send it to me at my floppy disk drive repair service.

For your information, we charge $25 for cleaning, lubrication, and complete alignment of any single-sided 35-, 40-, or 80-track drive. Double-sided drives cost $5 more. If something has to be fixed to achieve proper alignment, we charge $25 extra for troubleshooting and repair. That includes all parts except for heads, motors, and special LSI chips (found in units like Atari drives, for example). And if you need one of those noncovered parts, we'll give you the option of taking the unit back at no charge.

We service only 5¼-inch floppy drives but handle all brands, models, and configurations. We have a simple Atari 400 to test Atari-compatible drives, an Apple II+ to test Apple-compatible drives, and a VIC-20 to test Commodore drives. But you might be interested to know we use a pair of TRS-80 Model I computers with Percom Doublers to test all other drives, such as IBM, Texas Instruments, Osborne, Compaq, and Sanyo, after they come off the bench. *(Les Logan, Logan-Bower Mini-Floppy Center Inc., Norfolk, VA 23513)*

A: Thanks for your help.

Q: I have written several programs that require the deletion of records from direct-access files. I can put deleted records at the end of the file with keys such as ZZZZZ or something similar, but I would like to have the option to shorten the file length by changing the directory entry. Is there a patch or a POKE that changes the length characteristic in the directory of a TRSDOS 1.3 system so that I can shorten files? *(Richard Earp, Pensacola, FL)*

A: I know what you mean about wanting to delete such records, but fooling directly with the disk directory is not something you should do lightly. The DOS does more than just count the number of records in a file. It maintains a granule allocation table (GAT) that specifies which sections of the disk are free and which are occupied, a list of the tracks and sectors occupied on the disk by each file (in that file's directory entry), and the exact byte in the last sector immediately following the last byte of your file.

Changing the file length without changing the associated information in the GAT and directory entry is begging for a disaster. A time-consuming but simple way to delete such files is to write a Basic program that just copies the data to a new file that is the proper length, then deletes the old file and renames the new one with the old one's name. While this is slower, it has the advantage of letting the DOS do all the file location work, and doing it properly.

For more information about the design of the directory track, get Harv Pennington's book *TRS-80 Disk & Other Mysteries* from Montezuma Micro. The book was written for the Model I, but the directory track design is the same for the III.

Q: When I use a Model 4, an Epson FX-80 printer set for a 2K buffer, and Model 4 Basic, and I type in the command OUT 248,15, I may or may not get the desired compressed print. When it works correctly, exiting Basic leaves the printer in the compressed-print mode. What do I need to do to assure that the response will be compressed print? (I presume that what precedes that command is the key, and have tried preceding OUT 248, 15 with the command OUT 236, INP(252) OR 16.)

Also, how do you define drive 1 as logical drive 5? *(R.M. Doerr, Rolla, MO)*

A: What you are doing is sending the code 15 to your Epson printer. Another way to do the same thing is to type LPRINT CHR$(15) from Basic. I'm not sure why the Out command doesn't always work.

Setting drive 1 to drive 5 is simple. At the TRSDOS prompt type: SYSTEM (DRIVE = 5, DRIVER = "FLOPPY/DCT") and press enter. The floppy driver program will prompt you for the physical location of the drive you want readdressed. In this case type in "2" and press the enter key. And that's it. If you now type "DIR :5", the drive light on drive 1 will come on and the disk in it will have its directory scanned and displayed. Don't use this technique to make drive zero another logical drive. While you can do so, you might have difficulty trying to boot up your computer with the modified system.

Once you're satisfied with drive arrangement, use the SYSGEN command to save the new configuration to your disk. The next time you turn on the power, drive 1 will act as drive 5.

Q: In the August 1985 issue (p. 16), Carl Sturner wrote about a problem that he was having with SuperScripsit and the alignment for the special characters. Your suggestion to try different increments until he discovered the magic number was close to target.

I had the same problem and wrote to Tandy. They informed me that the spacing values listed in the Daisy Wheel Printer 410 manual were incorrect. If you print in elite or pica pitch, the width values are always 10 or 12 respectively. If you are printing in proportional spacing, the table values given in the printer manual on page 28 need to be multiplied by 2. After I followed these adjustments, my output lined up correctly. I didn't find anything wrong with SuperScripsit. *(David J. Kelton, Richmond, VA)*

A Thank you for informing us of the exact nature of the problem. And it's nice to know that the problem isn't SuperScripsit as we thought.

Q In response to Craig L. Cole's question in the February 1985 issue (p. 18), I have noticed one other upgrade for the Model I that seems promising. In the September 1984 issue (p. 182), Micro-Labs Inc. advertises 80-GRAFIX, a plug-in, clip-on board upgrade for any Model III/I to provide 128 user-definable characters. It comes with over 20 programs and costs $99.95. It's very brief and not well explained, but I would interpret the "user-definable characters" to be characters that use the "graphics" built in to the Model I: the 384 by 192 (I believe) pixels from which the computer creates the characters.

Is this board still made? If so, can you clarify what the board does and tell me how I can get it? *(Greg Bryant, Raleigh, NC)*

A Yes, it is still manufactured. You can order it from Micro-Labs Inc., 902 Pinecrest, Richardson, TX, 75080 (214-235-0915). Unfortunately, I don't have any more information than what was in that advertisement. If anyone out there has bought and used this device, would you like to tell us about it and give your opinions?

Q I have a Model I Level II computer. I bought the parts from Radio Shack and installed a lowercase kit without realizing that I need a driver program. Then Radio Shack told me that they could not furnish the driver! Can you or one of your readers help me on this one? *(Edward R. King, Bloomington, IL)*

A Dennis Kitsz's book, *The Custom TRS-80 & Other Mysteries*, has a short machine-language driver you can use either in DOS or Level II Basic, as well as a key repeat/debounce routine. This book is currently available from Montezuma Micro. If you don't already have it, you'll find it an excellent investment for your Model I. In addition, when you upgrade to DOS, you'll find that most DOSes automatically include an upper-/lowercase driver as part of the system.

Q I am acquiring a Model 4 and an MS-DOS machine. I'd like to keep my Model I on-line for communications, but it takes up a lot of space. The solution would be to hide the expansion interface and the central processing unit under my desk, extend the monitor cable, and then buy an external keyboard with a long cable. How could I patch the new keyboard into the system or where can I find information on same? *(Joel M. Reed, New York, NY)*

A Dennis Kitsz's book, *The Custom TRS-80 & Other Mysteries*, has just the solution you want. Kitsz designed a remote keyboard and video setup to let him put his Model I in one room while working in another (this was so he could sit beside his warm wood stove without worrying about the smoke or dust contaminating the computer or its drives). It isn't difficult; it just requires a little soldering work.

Q I am writing in response to Charles H. Samuel's question regarding the sort from the Tandy newsletter (June 1985, p. 17). The code in question is a call to the CINT function in ROM, CD 0A7F. This code is in the same address in both Models I and III and converts the number the USR statement passes to an integer in the HL register pair. Many machine-language programs use this call to properly load HL with the passed parameter.

The easiest way to implement these subroutines on the Model 4 is to replace CD 7F 0A with zeros (NOP instructions), then define a variable, such as Sort, as the starting address. If the integer variable N contains the number of elements, you can invoke the subroutine by the statement CALL SORT (N).

Model 4 Basic appears identical to MBasic in CP/M. The pointer to the variable in parentheses automatically loads into the HL pair. This is described in the TRSDOS 6 manual in Part II under the Call statement.

People accustomed to loading an integer array with multiple parameters can use this same method. Delete the CD 0A7F statements in the routines, define a variable to point at the entry point, and execute the calls by the Basic statement Call routine (P%(0)).

Not only does this work, but following the purpose of machine-language subroutine calls becomes easier, since you can make a variable name more descriptive than a USR statement. *(Larry E. Fosdick, Athens, GA)*

A Thanks for troubleshooting the sort routine. You can now use it on all the low-number series Tandy computers, Models I to 4.

Q I am writing about a letter from a reader in Germany (January 1984, p. 28) that described the problems he had with his computer when he tried to connect it to the 220-volt, 50-hertz power line. We have this kind of power line in Argentina and I recently had a similar problem with the drive motor self-starting.

After a long session with an oscilloscope checking the various test points in the computer, I found a problem in the power transformer, which has a primary winding designed for 110 V and 60 Hz. Apparently the transformer was designed with little margin for overload. When you connect it to a 50 Hz line, it overloads due to the overmagnetization, and the output results in a distorted sine wave with plenty of harmonics.

The drive, a Tandon TM-100 sold by Radio Shack as an external unit for the Model III, has two power supplies: a +5 V and a +12 V. The +5 V is built around a three-terminal regulator, which is a high gain device and subject to auto oscillations. Because the power supplies put out a distorted waveform, the harmonics reach sine wave proportions and the regulator would oscillate at regular intervals.

The control lines of the drive are active low (low voltage indicates a logical zero), so the Motor On line (among others) is pulled up to a logical 1 (+5 V) to signify an off state. Because the logical 1 depends on the +5 V line, when the regulator oscillates this power line drops to zero. The servo motor, which works from the +12 V line, is fooled into thinking a true Motor On signal has been received and turns on the drive motor.

The solution is to replace the current 110 V power supply with one using a 220 V 50 Hz primary winding. *(Javier Henderson, Buenos Aires, Argentina)*

A Thank you for a clear description of the mysterious overseas self-starting drive problem. If your overseas system suffers from this complaint, just replace the current drive power supply with a new one designed for the 220 V power grids frequently used worldwide.

Q This is a response to William Kirksey's question about how to transfer Radio Shack's MicroChess from tape to disk (August 1984, p. 14). I have a dual drive 48K Model III running TRSDOS 1.3. By using the Tape command I could transfer my version of MicroChess from tape to disk. Since you have a Model III, try using TRSDOS 1.3 to make the transfer. *(S.R. Perry, Hayward, CA)*

A So it is possible to move the game to disk. I was afraid that there might have been some noncontiguous code in it that precluded the transfer. Thanks.■

You can reach Radio Shack's National Parts Division at 900 E. Northside Drive, Fort Worth, TX 76102, 817-870-5662. M/C and Visa accepted; each order has $1.50 handling charge.

Terry Kepner is a freelance writer and programmer, and an associate editor of 80 Micro. He's been writing about microcomputers since 1979.

Attention TRS-80™ owners who now have a "PC"

SuperCross/XT

EASILY TRANSFER FILES *FROM* THE TRS-80™ *TO* MS-DOS™ OR CP/M™ *AND BACK!*

80-MICRO summed it up, "The value of this program far exceeds its price... if you work with different computers, it's a must."
4-STAR Review - July 1985 issue

Don't be confused by competitors that are really BASIC translators with very limited transfer capabilities!
SuperCross/XT and SuperCross/XT-Plus are the MOST powerful file-transfer utilities available for the TRS-80!

SuperCROSS/XT will allow you to COPY files back and forth between different operating systems. Up to 170 of them - including PC/MS-DOS 1.x,2.x/3.x (single or double-sided), CP/M+, or CP/M 2.2 on your TRS-80™ Model 4/4P, III, or I/DD. You can do this with your existing hardware and *SuperCROSS/XT*, eliminating modems, cables, and terminal program transfers. *SuperCROSS/XT runs as a /CMD file under your TRS-80 operating system.* Data files, spreadsheet files, and text files can also be moved between machines, like years of Visicalc™ files, business letters, legal drafts, or medical records, for example.

Comments and letters on-file from registered users are unusually enthusiastic about this product and its ease of use. *It WILL do what you think it will do, it's easy, and it WORKS GREAT!* New features in SuperCROSS/XT include "tagging" files for multiple **COPY**'s or **KILL**'s to eliminate *many* unnecessary keystrokes!

Some unsolicited customer comments from our registration cards...
"Powerful & easy to use" - GFP, Chicago IL
"A very handy product worth much more than the price." - KA, Port St. Lucie FL
"Superb product!" - JF, Calgary Canada
"Excellent product! Works as advertised." - GF, Boise ID
"Works! Rejoice! I can now talk to PC's!" - CN, Saratoga PA
"Excellent. No problems at all." - RH, Santo Domineo Dominican Republic
"Super! It performs the task I needed done." - TT, Rockville MD
"Fantastic! I transferred my files within 1 hour!" - PJS, Rolla MO
"LOVE IT!" - Col. CDL, APO San Fransisco CA
"Doc is straightforward & understandable. Solves my problem." - DG, C.Chase, MD
"Top Notch! Works for me." - DB, Lancaster PA
"Outstanding! Great product as usual." - CL, Los Angeles CA
"Will save 100's of hours! It'll protect my 4P from obsolescence" - RJ, Denver CO
"Program works very well. I like it" - MDM, Rochester, NY
"Superb. Easy to understand documentation." - RES, St. Louis, MO

CNVBASIC/CMD, available seperately, "preps" your BASIC programs before sending over with SuperCROSS/XT. It will make most of the syntax and spacing changes required for converting Model I/III BASIC programs for use on MS-BASIC, CP/M BASIC, or Model 4 BASIC. Complex or commercial business packages written in BASIC probably will not convert 100% over by our *or any other* BASIC translator.

***Some of the DOS Formats Supported:** PC/MS-DOS variations include 1.x, 2.x/3.x single or double-sided (IBM and most compatibles), and Tandy 2000. CP/M variations include most well known single and double-sided formats including TCP/M 3.0+, Montezuma Micro 2.2 (*all* versions), Holmes and up to 160 others including: ALTOS, CROMEMCO, DEC, EAGLE, EPSON, HP 125, CP/M 86, KAYPRO, LNW-80, MAX-80, MORROW, NEC, OSBORNE, OTRONA, SANYO, SUPERBRAIN, TELETEK, TELEVIDEO, TRS-80 LIFEBOAT/I, OMIKRON/I, HOLMES/3, HURRICANE COMPACTER/3, SHUFFLE BOARD/3, XEROX, ZENITH-HEATH, & 8" std. CP/M.

Important! **Please specify Mod I/DD, III, 4/4P, or Max-80 computer type required.**

SuperCross/XT (70 disk formats)*	$ 99.95
CNVBASIC/CMD alone	$ 29.95
**BOTH! • COMBO Special!*	$119.95
SuperCross/XT-*PLUS* (170 disk formats)	$129.95
**BOTH! • SuperCross/XT PLUS Combo Special!*	$149.95

*includes CNVBASIC/CMD

Super Utility
for PC/MS-DOS™

Fixed/Floppy Disk-Tools for IBM PC™, XT™, AT™, Tandy 1000, 1200, 2000 and other compatibles.

Data file recovery, disk exploring, zapping, and MUCH more!

Accidental deletions and disk directory problems *can happen to anyone at anytime*... a power spike, fingerprint, speck of dust, hardware problem, or simply typing DEL "." in the wrong sub-directory can destroy critical data in a moment *without warning*. Therefore we introduce *Super Utility* for the PC - an easy to use disk utility containing many functions sorely needed in today's PC/MS-DOS computing environment.

With ***SU/PC*** you can restore damaged or deleted files using two different methods (one very easy, the other a bit tougher). Even clusters of an erased file assigned to another file can still be restored, unless the user has physically written over every byte of the original data (most applicable with word processing files).

In addition to file repair and recovery, ***Super Utility*** provides sector verify, sector editing, modification of sectors in Hex or ASCII, ease of renaming of files and setting their attributes (lock and unlock files, etc.), string search, copying sectors to a file, diagnostic sector checking, mapping of the FAT table of a file or an entire drive, visual graphics pertaining to your system, and full directory and sub-directory editing without endless menu-hopping - all in one program. Want to change the name of your sub-directrory without copying all your files to a new one? Just retype a new name over the old one with *SU/PC!* The sector display mode displays all 512 bytes on-screen at one time and allows you to fully explore your disks. SEARCH and CHANGE are nice here too! Will find any occurance of a byte or string on your disk. Compatible with DOS versions 2.0 - 3.1 on most systems. Some computers may require the use of PC-DOS. Color, composite, or monochrome video are supported. A great tool for fixed disk users as well as floppy. Also compatible with IOMEGA's Bernoulli Box™ storage device (soon to be distributed by Tandy as the Disk Cartridge System). Easily the most *POWERFUL* PC/MS-DOS disk utility available today. Please do not confuse the MS-DOS version of ***Super Utility*** with the TRS-80™ version. ***Super Utility for MS-DOS is NOT SU+ from the TRS-80 "ported over". SU/PC*** is a totally new program that fills the gaps that PC users have most need of and have asked us for. It's aimed at the beginner, the "office user", hobbiests, students of the PC, or programmers alike. *Many of our users say this is much easier than NORTON's.*

So, why not be certain about the safety of your data (and your peace of mind) when you can have ***Super Utility/PC*** disk insurance right on hand at an unusually low price? Make your new computing life easier, more fun, and knowledgeable all at the same time. ***Unprotected.***
Requirements: IBM PC or compatible running PC/MS-DOS 2.x-3.x, minimum of 128K memory, and at least one disk drive. PC-DOS may be required for use.

IBM, PC, XT, & AT are registered trademarks of International Business Machines Corp. MS is a reg. trademark of Microsoft. TRS-80 is a reg. trademark of Tandy.

• Introductory Offer •

only $89.95

Prepaid or charge card orders (only) include free UPS shipping to US addresses! Blue label, COD, or over-night shipping available at extra charge. Visa-MasterCard accepted. Canada, please add $3; airmail. Other countries add $15 for airmail. Foreign orders, please use Charge Cards ONLY. Checks not drawn on US banks not accepted. ***Texas residents must add appropriate sales tax!***

POWERSOFT
17060 Dallas Parkway, Suite 114
Dallas, TX 75248
(214) 733-4475

Holiday Specials from PowerSoft

SUPER UTILITY Combo Specials

Attention *Super Utility* Users: *(registered or otherwise)*
Our ever-popular book *INSIDE SU+ 3.x* has been revised and renamed to now include all the changes that have occurred in the past two years since it was last revised. This perfect-bound, large format, slick 100 page+ book explains all the ins and outs of using these powerful utilities to the fullest of their capabilities. If you own *SU+ 3.x, SU4/4P*, or *PowerTOOL* this book will really add to your knowledge and maybe explain some things you didn't understand before. Lots of tips, hints, and suggestions are included, as well as helpful information for the novice as well as the "pro". Disk theory is explained as well. The book is now shipping, and a real value at only $19.95, its list price, but this month we're offering it for only $15! *Save $5!* If you are a new Model 4 owner and a new *SU4* owner as well, and never bought the earlier edition of this book, then *NOW* is the time to get some great reading material about what you have!

Don't own *Super Utility* yet?? Buy our world-famous *SUPER UTILITY+ 3.2* or *SUPER UTILITY 4/4P* and add our new addition of *USING SUPER UTILITY* for only $5 extra! *Save $15!*

Super Utility+ 3.2 for the TRS-80™ I, III, 4(III)	$79.95
Super Utility 4 for the TRS-80™ Model 4/4P/4D	$79.95
USING SUPER UTILITY Book Sale price	$15.00

>> Special combo SU with new book for only $84.95! <<
Super Utility for the TRS-80 is "protected". It comes with two copies of the program, however. Upon registering, you may order an unprotected copy

*****

PRONTO from MISOSYS - a *Sidekick* type program for the Model 4! *YEA!!* Memory resident utilities including calculator, name/address file, phone numbers, notes, and more! All available at the stroke of a few keys! Retails for $59.95, but we bought a load of these *(it's really great!)* and want to pass them to you at the **special holiday price of only $49.95.** ***SAVE $10!***

TRSDOS 6™ Programmer's Guide - only $15.00
Excellent book by Roy Soltoff, contains many goodies left out of Tandy's *Model 4 Technical Reference Manual!* Not for beginners.

*****

Complete Automated Communications Setup for the TRS-80 Mod I, III, or 4(III)

Outgoing and incoming as well!

The ST80-III Smart Term Communication System by Lance Micklus is now available from PowerSoft! ST80-III v3.5, an award winner, was formerly $150 by itself, the **X-10 HOST** package; which allows you to leave your computer "on-line" securely with full password protection, etc. (formerly $50), and Lance's **Personal Bulletin Board System** (formerly $40) all together now in one package ***for the special price of only $69.95! You save OVER $100!*** If you have a modem (especially a Hayes or other auto-answer type) and don't have good software, *this is the one to get! Now get EVERYTHING, in the way of telcom software, for your computer at one low price!* UNPROTECTED MEDIA. Requires only a single disk drive and 32k of memory. Please specify Model I or III/4(III mode).

Only $69.95 complete!

*****

Do you use and like Scripsit™, but wish it did a WHOLE LOT MORE? **You'd like a lot of new features without relearning a whole new system?** *Our PowerScript 4.2 is just the answer! PowerScript 4.2* adds two categories of additions to SCRIPSIT™. Directory/File functions (DIR, FREE, KILL CHAIN, LINK), and embedded printer control functions. With all the smarter printers out there, it doesn't make any sense to use a "dumb" word-processor anymore. In this case, "dumb" means it can't really control your printer to its fullest capacity. If you notice, there are MANY word processors out there for the TRS-80™ besides the ones from Tandy. We could recommend any one of these, *IF YOU WANTED TO BUY SOMETHING NEW.* One of the problems, however, is that you need to learn all new commands. The second problem (maybe the main one) is that you need to shell out a lot more money to buy it! *PowerSCRIPT 4.2* gives you many new features, while retaining the commands you already know - at a very low price of only ***$39.95!*** All your previous files will, of course, be compatible. *PowerScript 4.2* received a 4 1/2 star rating in *80-MICRO*'s July '85 issue. *Supports Mod I, III, and 4 versions of SCRIPSIT™.*

Add all this power to your program for only $39.95!

*****

Do you have a hard drive on your TRS-80?

PowerSOFT has a complete line of *Supreme Hard Disk Driver* packages that offer *much greater flexibility* and ***superior performance*** over the standard drivers supplied by Tandy or other hard drive manufacturers for the TRS-80 Mod 4, III, or I. They allow you to split up your hard drive into partitions of differing sizes, and also allow you to define granule sizes and other optimizing operating characteristics to achieve maximum from your drive. The drivers themselves can be used on hard drives of various capacities, *not* just 5 megabytes, but any of the popular larger sizes now being sold. At the same time, the driver routines themselves (once relocated) are very small (usually less than 256 bytes. Most hard drive companies either sell or recommend our drivers. These drivers can also be used to allow a hard drive to be shared between LDOS 5.1 and TRSDOS™ 6 for Model 4 or 4P owners! You can keep all your Model III programs AND your Model 4 programs, as well as the data, all on one hard drive. Just by booting a different disk (LDOS or TRSDOS 6) you are in the Mod III or Mod 4 mode running off your hard disk in either case! It's great. No reason to run from floppies when you can use the hard drive in either mode! If you have the 15 meg or 35 meg Tandy drive, you can now format it for full useage - NOT just 5 meg (Mod I, III or 4 mode)! If you have a 4P, you can boot directly off the Tandy hard drive without a disk! Really! Just turn on the power. We have drivers for Tandy, Percom/Aerocomp, Hard Drive Specialists, and other hard drives using a Western Digital WD-1000 or compatible controller board. Call us for complete details or write for a spec sheet. *Only $99.95 complete.*

Whether you use our software driver or not, you NEED to use *BACK/REST! **See the 5-Star review in the Oct'85 issue.** Shame on you if you're not backing up your data!* If you've invested good money into a hard drive system, *it doesn't make any sense not to have a BACKUP routine.* BACK/REST makes that job easier and saves *much* time. And TIME IS MONEY. ***Don't delay! Save HOURS!*** BACK/REST will work with all that use LDOS or TRSDOS™ 6 *including Tandy 5, 15 and 35 Meg drives.*

Hard Drive COMBO Specials!

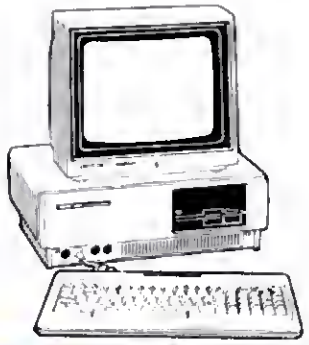
BACK/REST 1.3 *for TRSDOS 6™ & LDOS*		$99.95
BACK/REST *for DOSPLUS 3.5*		$75.00
LDOS ToolBox *Hard Disk Repair & Recovery Tools for LDOS - Mod I/III*		$49.95
Model 4 ToolBelt *Hard Disk Repair & Recover Tools for TRSDOS 6 - Mod 4, 4P, 4D or if your hard drive is split Mod 4 and III - get both!*		$49.95
Both sets of tools for only		$75.00!
BACK/REST **and your choice of** ***TOOLBOX for LDOS*** **or** ***MOD 4 TOOLBELT***	*only*	$129.95!
BACK/REST **and BOTH sets of tools**	*only*	$159.95!
BACK/REST **and our** *Supreme Rigid Driver* *	*only*	$149.95!

*add either *LDOS TOOLBOX* or *Model 4 ToolBelt* for only $29 more!

Fine print: These prices are for prepaid orders only. School purchase orders will not be accepted at these prices. Visa or MasterCard gladly accepted. COD orders will have $3 added to the total. Flat rate shipping on any item or combination of items from this ad is only $3. Blue Label or overnight available at extra cost. Canadian airmail is $4 and other foreign airmail will be actual cost. Checks not drawn on U.S. banks not accepted. *Texas residents must add appropriate sales tax.*

POWERSOFT
17060 Dallas Parkway, Suite 114
Dallas, TX 75248
(214) 733-4475

PERRY COMPUTERS



CALL
1-800-248-3823

COLOR COMPUTERS		LIST PRICE	OUR PRICE
26-3127	64K Extended Color Computer 2	$219.95	$175.00
26-3134	16K Standard Color Computer 2	119.95	99.00
26-3129	Thinline Disk Drive O for Color Comp	349.95	290.00
26-3018	Extended Basic Kit	39.95	36.00
26-3030	OS-9 with Editor Assembler	69.95	59.50
26-3012	Deluxe Joystick	29.95	25.00
26-1208	CCR-81 Recorder	59.95	50.00
	VIP Integrated Library	149.95	139.00
	VIP Writer	69.95	59.00
	VIP Calc	69.95	59.00
	Telewriter Disk		59.00
	Botek Interface		57.00
TANDY 2000			
26-5103	Tandy 2000 Two Disk	$1599.00	$1225.00
26-5104	Tandy 2000 One Disk & 10 Meg HD	2499.00	1885.00
26-5111	Monochrome Monitor	199.95	165.00
26-5112	Color Monitor	599.95	510.00
26-5140	Hi Res Graphic Monochrome Board	299.95	255.00
26-5141	Hi Rec Color Graphics Chips	124.95	105.00
26-5160	Internal 128K Memory Board	179.00	150.00
26-5300	Lotus 123	495.00	420.00
26-5311	Microsoft Multiplan	195.00	165.00
26-5320	Framework	695.00	590.00
26-5352	dBase II (Data Base)	595.00	420.00
26-5330	Multmate	249.00	205.00
MODEL 200 COMPUTERS			
26-3860	24K Model 200 Portable Comp.	$999.00	$725.00
26-3866	24K RAM Mem. Exp Chip for 200	249.95	210.00
26-3804	AC Adaptor	5.95	5.00
26-3805	Accoustic Coupler	39.95	34.00
26-3816	8K RAM Memory Expansion Chip	119.95	95.00
26-1409	Printer Cable	14.95	12.70
26-1410	Modem Cable	19.95	17.00
26-3809	Briefcase	49.95	42.50
26-3811	Soft Carrying Case for 100/200	39.95	34.00
26-1183	Bar Code Reader	99.95	85.00
26-3806	Disk Video Interface	499.00	415.00
26-3829	Multiplan ROM for 100	149.95	127.50
MODEL 4 COMPUTERS			
26-1070	Model 4D 64K 2 Drive Computer	$1299.00	$ 895.00
26-1122	64K Memory Expansion	69.95	65.00
26-1127	Model 4 Internal Drive 1	399.95	385.00
26-1123	Model 3 to Model 4 Upgrade Kit	399.00	385.00
26-1084	Model 4P Modem Board	149.00	95.00
26-1085	Model 4P Travel Case	24.95	21.25
26-1134	Model 4 Hard Disk Kit for 15meg	79.95	67.95
26-1530	Multiplan Model 4	199.00	169.00
26-1595	Super Scripsit Model 4	199.95	169.00
26-1635	Profile 4 Model 4	249.95	200.00
26-2216	CP/M Plus Model 4	149.00	127.00
26-2231	Double Duty Utility	69.95	58.00
MODEL 12 & 6000 COMPUTERS			
26-6021	Model 6000 512K 2 Drive Comp	4499.00	3125.00
26-6022	Model 6000 512K 15Meg HD	5499.00	3795.00
26-6014	Model 12 to 6000 Upgrade	1595.00	1325.00
26-6015	Model 6000 512K Memory Board	1095.00	905.00
26-6019	Model 6000 256K Memory Kit	199.95	185.00
26-6017	Card Cage for Model 12	199.00	169.00
26-6052	DT-100 Data Terminal	795.00	600.00
26-4155	15 Meg Hard Drive Primary	1995.00	1355.00
26-4171	35 Meg Hard Drive Primary	2995.00	2095.00
26-4157	Installation Kit for Primary HD	349.00	265.00

All prices and offers may be changed or withdrawn without notice. Advertised prices are cash prices C.O.D. accepted ($10.00 charge per carton on C.O.D. Call for further C.O.D. information.) M.C., Visa, AX, add 2%. All non-defective items returned will be subject to 10% restocking fee. Defective items require return merchandise authorization. Call for R.M.A. Number before returning. Delivery is subject to product availability.

RADIO SHACK PRINTERS		LIST PRICE	OUR PRICE
26-1276	DMP-105 Dot Matrix Printer	$ 199.00	$ 169.00
26-1275	TRP-100 Portable Thermal Printer	299.00	230.00
26-1280	DMP-130 100 cps Tri. Mode Printer	349.00	285.00
26-1268	CGP-220 Color Ink-Jet Printer	699.00	595.00
25-1257	DWP-220 Daisy Wheel	599.00	460.00
26-1277	DMP-430 24 Wire Matrix Printer	899.00	745.00
26-1270	DWP-510 43 cps Daisy Wheel	1495.00	1225.00
26-1274	DMP-2100P 24 Dot Wire Matrix	1995.00	1650.00
26-1279	DMP-2200 HiSpeed Matrix Printer	1695.00	1440.00
26-1269	PT-64 Printer Controller	249.95	210.00
26-1498	SW-302 Printer Switch	119.95	100.00
26-1477	Auto Sheet Feeder for DWP 510	499.95	420.00
TANDY 1200			
25-3000	Tandy 1200 Dne Disk & 10 Meg HD	$1999.95	$1525.00
25-3010	VM-3 Green Monitor	219.95	185.00
26-3212	CM-2 Color Monitor	459.95	390.00
25-3040	Monochrome Display Adapter	219.95	185.00
25-3043	Graphics Display Adapter	299.95	255.00
25-3044	Graphics Master	695.95	540.00
25-3061	Captain Multifunction Board	599.95	475.00
25-3130	M5DOS/BASIC	89.95	76.50
25-3170	Wordstar Professional	395.95	335.00
25-3160	Framework	695.95	590.00
25-3161	PFS File	140.95	119.00
25-3190	dBase III	695.95	590.00
TANDY 1000			
25-1000	Model 1000 128K 1 Disk Drive	$ 999.95	$ 705.00
25-1001	Model 1000 256K 10Meg HD	1999.00	1495.00
25-1005	Disk Drive Expansion 1000	199.95	170.00
25-1006	RS-232C Interface 1000/1200	99.95	85.00
25-1007	Hard Disk Control Board	299.95	255.00
25-1011	Memory Plus Board	319.95	255.00
25-1013	Modem 1200 Option 1000/1200	299.95	230.00
25-1025	10 Meg Hard Disk	699.95	559.00
25-1501	M5-DOS Reference T-1000	34.95	29.00
25-1502	BASIC Reference T-1000	34.95	29.00
26-3211	VM2 Green Monitor	149.95	125.00
26-3212	CM2 Color Monitor	459.95	390.00
EPSON/COMREX PRINTERS			
20001015	Homewriter 10	$ 288.00	$ 230.00
20001025	LX-80 Dot Matrix Printer	299.00	250.00
20001035	FX-85 Dot Matrix Printer	499.00	385.00
20001040	JX-80 Color Dot Matrix Printer	699.00	485.00
20001050	HI-80 4 Pin Plotter	599.00	390.00
20002010	RX-100 Dot Matrix Printer	499.00	400.00
20002030	FX-185 Dot Matrix Printer	699.00	525.00
20003010	LQ-1500 18 Pin Head with Intf	1495.00	1095.00
20003510	CR-IIE Comrex Com Riter	499.00	295.00
20001510	FX-80 Tractor Feed	39.95	34.00
20004080	NLQ Parallel Intr	209.95	175.00
20001515	LX-80 Tractor Feed	49.00	40.00

For Technical Questions and Information on our complete line of computer accessories and current prices.

CALL 1-517-625-4161
FOR ORDERS ONLY CALL 1-800-248-3823

Mon., Wed. & Fri. 9-9, Tues. & Thurs. 9-6, Sat. 9-3

124 S. MAIN ST, PERRY, MICH. 48872

Tandy's 1985 Seesaw: Revenues Up, Profits Down

Tandyland

Financially speaking, last year was pretty tough for Tandy. Although the company took in 2 percent more money in fiscal 1985 (which ended June 30) than in 1984, its net profit dropped almost 33 percent, the first such drop since 1978 (see the Figure).

In raw figures, Tandy reported a $189.1 million profit on sales of $2.84 billion for fiscal 1985. That compares with a $281.9 million profit on $2.78 billion in sales the previous year.

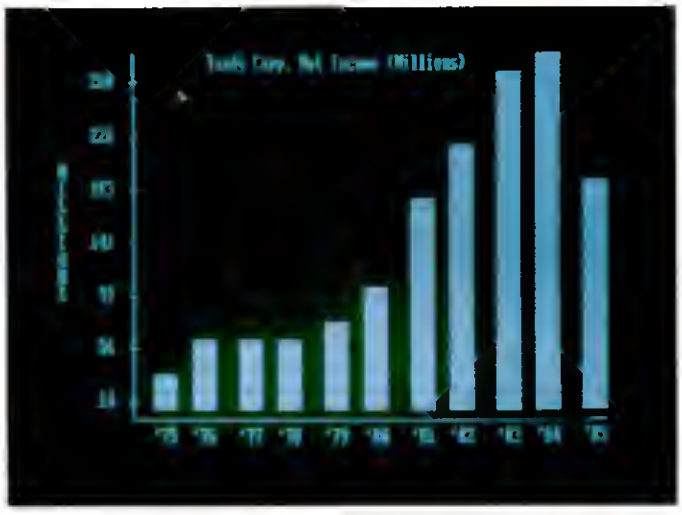


Figure. Tandy's annual net profits for 1975—1985.

In comments published in the *Fort Worth Star-Telegram*, Garland Asher, Tandy director of financial planning, blamed Tandy's bad year on the slumping business computer market. Asher cited as evidence disappointing sales of the Model 2000, on which Tandy took an $18 million write-off last April (see Pulse Train, August 1985, p. 21). The success of the Tandy 1000 and strong sales of the 1200 HD further weakened the 2000's sales position, according to Asher, and Tandy accordingly cut the 2000's price to $1,599 in July.

Despite last year's financial setbacks, both Tandy officials and industry analysts remain upbeat about the company's future. Don F. Sinsabaugh of Swergold Chefitz, a New York investment banking firm, sees some exciting new products on the horizon for Tandy. However, none of them is in the microcomputer market. "The videocassette recorder market will continue to grow. In audio, compact discs are strong, and cellular communications will have strong growth in the next couple of years as prices come down. Tandy will get its fair share of that." Meanwhile, Tandy's Asher says, "This is going to be a big year for new products, both in the computer area and (for) other merchandise."

So far, the optimism seems justified. July 1985 sales indicate a rebound from Tandy's dismal fiscal 1985 numbers: Worldwide sales were up 12 percent over July of last year, and U.S. sales were up 19 percent.

With January approaching, we'll soon see if the rumors of a new Tandy Color Computer are on target. Speculation about a new breed of Color Computer has gone on for two years now, but recently the rumors have been full of explicit details.

Word is that the machine will run under Microware's OS-9 operating system with a 640- by 400-pixel screen, up to 512K of RAM, and one double-sided 3½-inch disk drive. The scuttlebutt on CompuServe puts the computer's price at $499.

For old CoCo users, the good news is that Tandy will still sell the CoCo II, but at the reduced price of $99 and only in a 64K configuration. Tandy might bundle DeskMate with the new computer, as they're doing with the Model 4D and the Tandy 1000. The CoCo DeskMate costs $99.95 and has two more functions than the 1000 and 4D versions: a simple paint module and a general-ledger module.

As usual, Tandy won't confirm or deny reports of a more powerful CoCo in the offing.

Meanwhile, speculation about a new Tandy laptop can, for the moment, be put to rest. The anticipated Tandy 600, which, according to rumor, would take on Data General's DG-1 MS-DOS portable, is vaporware.

According to my sources, Tandy did develop a new laptop, though it wasn't MS-DOS-based. But the feeling in the Tandy Towers was that the machine was a kludge and would never make it to market.

On the other portable hand, a new version of the Model 100 might show up soon. Tandy reportedly had a new thin-line 100 in the works as of early September, readying it for an October introduction. The revamped machine will have a minimum configuration of 24K RAM and should cost $299.

My source tells me that Tandy won't adapt the Model 200 to the new, thin design. While that might sound like a non-story, the reason behind the decision should spark some interest.

Apparently, Tandy's agreement with Kyocera Ltd. of Japan, makers of the 200, was for a limited order of 85,000 units. The stipulation was that if the machine didn't sell as well as expected, Kyocera would produce no more machines, and Tandy would simply sell off its stock until it was gone. At this point, Tandy's still working off the original lot of 85,000 computers, and has no plans to order any more.

I often report how Tandy computers do against their competitors, but it's worth mentioning that Tandy also sells a significant number of printers, monitors, disk drives, and modems. In its July 22, 1985, issue, *Computer + Software News* published June sales figures for computers and peripherals, and Tandy ranked no lower than second in any category (see Table 1).

Of the best-selling personal computer brands, Tandy/Radio Shack placed second with 20 percent of the retail market. They were second in the printer category with 18 percent of the market, and first in sales of monitors and disk drives, with shares of 18 and 21 percent, respec-

Photograph by Frank Cordelle

	Brand	% retail sales
Computers	IBM	30
	Radio Shack	20
	Apple	19
Printers	Epson	18
	Tandy	18
	Apple	15
Monitors	Tandy	18
	Apple	14
	IBM	14
Disk Drives	Tandy	21
	Apple	19
	IBM	7
Modems	Hayes	43
	Tandy	20
	Apple	13

Table 1. Best-selling brands of personal computers and peripherals during June 1985.

tively. Tandy's share of the modem market was 20 percent, good enough for a second-place finish.

MicroTrends

Some microcomputer companies aren't happy about a recent deal between IBM and the Mexican government, whereby IBM will own and operate a microcomputer manufacturing plant in Mexico. That's a significant departure from Mexico's national trade law requiring that computer plants have Mexican majority ownership.

Earlier this year, Mexico let Tandy start manufacturing Model 1000 computers in a Mexico City plant, but limited Tandy's ownership stake to 49 percent. Similarly, Apple has a minority share in an operation that produces Apple IIs in Mexico.

Tandy's reaction to the exception was subdued; they were generally pleased to be able to produce their top seller in Mexico and open up a Latin American distribution network for the 1000. But Richard Hojel, chairman of Apple de Mexico, didn't like it a bit. "What we're seeing here is a tremendous amount of armtwisting by a very powerful company," said Hojel. "In principle I'm in complete agreement with IBM's presence, because I believe the best defense of private enterprise is competition. But let's all play by the same rules."

In August, Microsoft and IBM penned a joint software development agreement that virtually guarantees Microsoft's position as developer of future IBM PC operating systems. The move quashed rumors that IBM was preparing to introduce a proprietary operating system for its PC line. In an interview with CW Newsnet, IBM analyst Michele Preston of L.F. Rothschtld, Unterberg, and Towbin said, "The agreement puts to rest whatever questions remained about IBM moving away from DOS. It's very positive for the industry." Microsoft is apparently free to license jointly developed products to other manufacturers, good news for Tandy and other makers of IBM compatibles.

Device	% sold 1983	% sold 1989 (est.)
Digitizer	33.2	18.6
Data tablet	11.9	36.4
Light pen	9.0	4.4
Touch screen	6.4	12.2
Joystick	17.5	5.1
Trackball	6.8	3.3
Mouse	9.4	13.2
Speech	5.8	6.8
Total Sales	$131 million	$962 million (est.)

Table 2. The U.S. workstation interface device market.

Market researchers at International Data Corp. see a bright future for companies manufacturing data entry devices such as data tablets, light pens, and touch screens. IDC expects sales of these units, collectively called workstation interface devices (WIDs), to increase sevenfold through 1989 (see Table 2).

In 1983, manufacturers shipped 1.6 million WIDs. IDC estimates shipments of almost 12 million units by 1989. In terms of revenue, IDC expects the WID market to grow from $131 million in 1983 to $1 billion in 1989.

Hot Items

GTE's putting their Telenet network in the hands of the everyday telecommuter with a service called PC Pursuit.

Subscribers can call all BBSes and online data bases within GTE's 12-city network for $25 a month, regardless of the amount of time spent on-line.

The catch is that you must also place your call from an area code covered by the network, which links Atlanta, Boston, Chicago, Dallas, Denver, Detroit, Houston, Los Angeles, New York, Philadelphia, San Francisco, and Washington. According to Telenet president David Hann, GTE chose those cities for start-up because their local calling areas contain about 23 percent of the U.S. population.

PC Pursuit supports 300-, 1,200-, and 2,400-baud operation; you can get more information about the system by calling 800-835-3001.

What would J. Edgar Hoover think? Microcomputer users who subscribe to CompuServe can now help track down fugitives on the FBI's 10 Most Wanted list.

CompuServe members can access a file provided by the FBI of biographical information on the reprobates. If you have the proper hardware and CompuServe's Professional Connection or Vidtex software, you can even get a high-resolution picture of the varmint.

To access the 10 Most Wanted list, type in GO FBI at CompuServe's ! prompt. You then choose a vagabond's name from a list displayed on the screen.

The reason for this service? According to CompuServe's Richard A. Baker, "Many of our subscribers are professionals such as doctors, lawyers, and dentists. Like everyone else, fugitives require the use of [professional] services. In addition, many of these fugitives have distinct scars, tattoos, and limps so alert subscribers may spot one of them."

Appealing to the public seems to pay off for the FBI. Since 1950, when the 10 Most Wanted program began displaying photos in post offices, citizen cooperation has resulted in the capture of 109 of the 366 fugitives on the list.

Update

Portable computers' popularity problems aren't confined to the U.S., according to Systems Concepts Ltd., a London-based research firm. In a study of the European portable market, they found that only 85,000 units sold on the continent last year, even though 12 million Europeans travel on the job.

Systems Concepts believes the market isn't understood yet. Instead of focusing merely on portables' size, they say, sellers should bill their products as "personal support systems." The report notes that buyers want more than a machine that fills out forms; they want a system that will improve the way they work. Specialized software for portables is another need, according to the study.■

TRS-80 AND IBM-PC Word Processing Program

Computers Supported
IBM-PC
IBM-XT
Tandy 1000
Tandy 1200
Tandy 2000
Model 4
Model 2
Model 12
Model 16
Modem 4 CP/M
Model 2 CP/M
Max-80
LNW

"For $129.95 you'll be hard-pressed to find a better overall word processing product."
(LSI JOURNAL, January, 1984)

- ☐ **80 × 24 DISPLAY** using TRSDOS 6 or any **MODEL III DDS.**
- ☐ Built-In **FORM LETTER** and **MAIL-MERGE** capabilities.
- ☐ Build-In **HELP SCREENS.**
- ☐ **55** seperately programmable **MACRO FUNCTION KEYS.**
- ☐ Performs **PROPORTIONAL-SPACE** right-margin justification on over 120 different printers (all drivers included **FREE**).
- ☐ Integrates with ***ELECTRIC WEBSTER*** spelling checker.
- ☐ Keyboard entry and printing of **31 European Characters** and special symbols.
- ☐ **IBM-PC, XT, Tandy-1000, Tandy-1200, Tandy-2000** versions for color or Monochroma, now only $199.95.
- ☐ **MODEL 2/12/16** varsions now available for **TRSDOS 2.0, TRSDOS 4.2 and CP/M. $199.95.**
- ☐ **LeScript** also available to run on **TRS-80 MODEL I, MODEL III, LNW-80, LNW-TEAM (80 × 24), Holmes VID-80 (80 × 24),** and **Lobo MAX-80 (80 × 24). $129.95**

CALL 1-305-259-9397

ANITEK

ANITEK SOFTWARE PRODUCTS
P.O. BOX 361136 • MELBOURNE FL 32936

FREE SHIPPING WITNIN THE U.S.; OUTSIDE THE U.S. ADD $10.00 FOR SHIPPING. FLORIDA RESIDENTS ADD 5% SALES TAX. ALL ORDERS PREPAID BY CHECK, MONEY ORDER, CREDIT CARD OR C.O.D.

Circle 141 on Reader Service card.

FREE

TRIAL-SIZE DISK OFFER

Fill out and return this coupon to receive a free trial-size copy of the **LeScript** word processing system - a $25.00 value.

Name _____

Address _____

City _____ State _____ Zip _____

Telephone _____

My system is

☐ Model I/III	☐ Holmes VID-80	☐ CP/M LNW-TEAM
☐ Model 4	☐ Model II	☐ CP/M VID-80
☐ MAX-80	☐ CP/M Model 4	☐ TANDY-2000
☐ LNW-TEAM	☐ CP/M MAX-80	☐ IBM PC

Send to ANITEK • P.O. Box 361136 • Melbourne, FL 32936

$99 ON SALE NOW

You've been seeing this ad for years.
Now come join us in the fun!
The PRODUCER is three years old
and we're
throwing
a spectacular
half price
birthday party!

*

Call Orders
Toll Free
1-800-433-5355

Add $5 shipping
$20 International

SPECIAL!
The Producer & Data Shuffler
just $125 When purchased together.
Save $110!

ANNIVERSARY SALE

Biasts from the Past

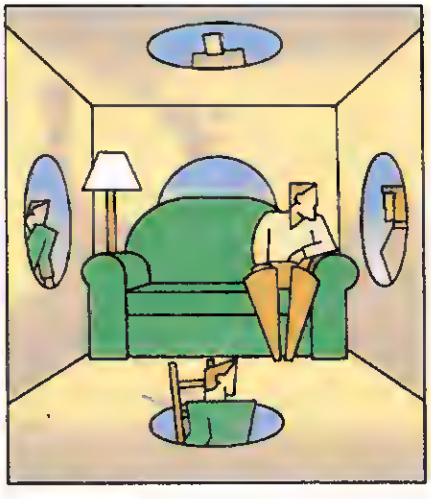
If you're using Arnold van Beverhoudt's **Graph Master program** (February 1985, p. 68) **with a DMP-120**, here's a tip from Alfred Kohlberg Jr. of New Carrollton, MD. Set DIP switch 1 to on and rewrite line 3038 to read LPRINT CHR$(10);:LPRINT CHR$(13);:Y = Y + 1: IF Y = 48 GOTO 3046.

Ray Pelzer tells us that his **Croascheck program** (September 1985, p. 66) **won't recognize the period aa a variable character** in a Basic program. Clifford 1. Knight cites the period as an undocumented but valid character in his article "Summer Romance: Learning to Love Model 4 Basic" (August 1985, p. 38).

Speaking of Cliff Knight, his **ScripAid modification** to Scripsit (January 1985, p. 60) apparently **ian't compatibic with Scripsit 01.00.01.** Don Coffin of Los Alamos, NM, found this out when he tried to use ScripAid with a version of Scripsit he had upgraded; the print functions dldn't work. If you've had the same problem, try using Scripsit 01.00.00.

We're still getting **Modei 4 scroll-protect routines** in response to our article "Stadonary Department" (May 1985, p. 74). The latest is from Andy Levinson: 10 NN% = 0:N%(0) = 78:N%(1) = 1798:N%(2) = 3902:N%(3) = 13841:NN% = VARPTR(N%(0)):CALL NN%(NL%):RETURN. Simply set NL% for the number of lines you want to protect. Program Listing 1 shows the source code.

Gilbert A. Emmert of Madison, Wl, submits a **modification to our FastBas Basic compiler** (January 1985, p. 42) that lets you specify an upper limit on the section of RAM FastBas uses and changes the reserved memory size from within the compiler. It also lets you determine the amount of variable space to set aside. You can now use compiled programs with other high-memory programs and compile machine-language subroutines more readily. Program Listing 2 lists the lines you should change. Also, delete line 1010. Finally, line 7275 determines HIGH$ in LDOS; other DOSes might require different addresses. If your DOS has no equivalent to HIGH$, delete line 7275.



Hints and Tips

Two readers have written in with ways to **activate the Modei 4P'a RAM tcst.** Bernard P. Tiltges of Lexington Park, MD, found that you can press the hyphen, left arrow, and right arrow keys simultaneously, while J.A. Kempen of Coevorden, Netherlands, discovered the 6-8-0 combination.

Michael Friedland of San Bernardino, CA, has a simple JCL file (Fig. 1) that lets you **aend printer controi charactera to an Epaon** from DOS, Basic, and some programs. You use the @ key followed by a letter. The @ key translates into an escape code. Thus, the printer reads @E as "escape-E."

Program Listing 1. Source code for scroll protect routine.

```
LD      C,(BL)          ;Get low byte integer argument from Basic
LD      B,7             ;Condition code for scroll protection
LD      A,15            ;VDCTL SVC code
RST     28B             ;Do the SVC
RET                     ;Return to Basic
```

Program Listing 2. FastBas modification.

```
512 PDKEM,P:PRINTP;:M=M+1:IFM<-12+TP THEN RETURN ELSE
CLS:PRINT:PRINT "Program has exceeded protected memory size"
1001 GOTO 720B
1005 Q=PEEK(16548)+256*PEEK(16549):L=1:K=0:FP=0:CP=0:MC=PEEK(16561)
+PEEK(16562)*256+3-65536:M=MC
1013 Q=Q1
1015
VT=-2*26+TP:VF=-4*26*(1+IS)+VT:VA=-4*NO*DO+VF:VD=-4*NT*DT*DT-2*
NT*DT+VA:VS=-NS*(SL+1)+VD:VN=-(SL+1)+VS:PRINT:PRINT "Zero
variables":GDSUB 7136 'CLEAR
7136 Cl=VN:GOSUB 814:P=175:GOSUB 512:GDSUB 902:P=119:GOSUB
512:Cl=VN+1:GOSUB 814:GDSUB 900:Cl=TP-VN-1:GOSUB 814:P=1:GOSUB
512:P=El:GDSUB 512:P=Dl:GOSUB 512:P=237:GOSUB 512:P=176:GOSDB
512:RETURN 'CLEAR
7200 IS=10:DO=20:DT=20:SL=40:NO=26:NT=2:NS=26
7210 PRINT "Number of additional S.P. variables per letter=";
IS;:INPUT"New=";IS
7220 PRINT"Dimension of 1-D arrays=";DD;:INPUT"  New=";DO
7230 PRINT"Dimension of 2-D arrays=";DT;:INPUT"  New=";DT
7240 PRINT"Length of strings=";SL;:INPUT"  New=";SL
7250 PRINT"Number of 1-D arrays allowed=";ND;:INPUT" New=";NO
7260 PRINT"Number of 2-D arrays allowed=";NT;:INPDT"  New=";NT
727B PRINT"Number of strings allowed=";NS;:INPUT"  New=";NS
7271 PRINT:PRINT
7275 BP!=PEEK(&B441l)+256*PEEK(&B4412):PRINT"BIG8$=";BP!;
7277 8P!=PEEK(16561)+256*PEEK(16562)+1:PRINT"  Start of reserved
memory";BP!
727B INPUT"New start of reserved memory=";BP!:BP!=8P!-1
7279 Dl=INT(BP!/256):El=BP!-256*Dl:POKE 16562,Dl:POKE 16561,El
7200 INPUT"Top of usable memory";TP!:IF TP!>BP! TBEN 72BB
72B4 IF TP!<BP! TBEN 7270
729B IF TP!>32767 THEN TP!=TP!-65536
7300 TP=INT(TP!):PRINT:GOTO 1005
7400 ' ************* END DF TRE COMPILER ****************
```

Illustration by Mark Fisher

Christmas SPECIAL OFFER !

FREE GIFTS FOR YOUR TRS-80

NEW LAZY WRITER

Lazy Writer

Lazy Writer, the innovative word processor, is NEW and BETTER THAN EVER!

- Customize your copy to work easily with any printer - supports printer special modes and typesizes.
- DOS error recovery - never face losing a whole file because of a bad file sector.
- Lazy Writer (Model 4 version) accepts only correct file names and warns you if the name you choose is already in use on your disk - you will never accidently overwrite a file you still want.
- Lazy Writer's famous two mode operation - Text Entry and Editing - makes use fast and easy.
- Edit with one-key mnemonic commands - "d" for delete, "i" for insert, etc. Same ease-of-use as always.
- Format your file before printing - see all page breaks, get an accurate word count.
- Start using it right away - a "welcome" program makes your working disk!
- Comes on mini MULTIDOS or transfers to the DOS of your choice.

LAZY WRITER FOR MODEL I, III, OR 4 ..... $124.95

LazyMerge Form Letter program will pull names from a mail list kept in Lazy Writer. No need to buy an extra mail list program, plus you get great flexibility. Insert any information from your mail list (including user defined fields) into a form letter, use all the formatting features of LazyWriter, including bold, underlining, etc. in your form letter.

BUY LAZY WRITER
GET LAZYMERGE
FORM LETTER PROGRAM FREE!

A $45 value
- your FREE gift

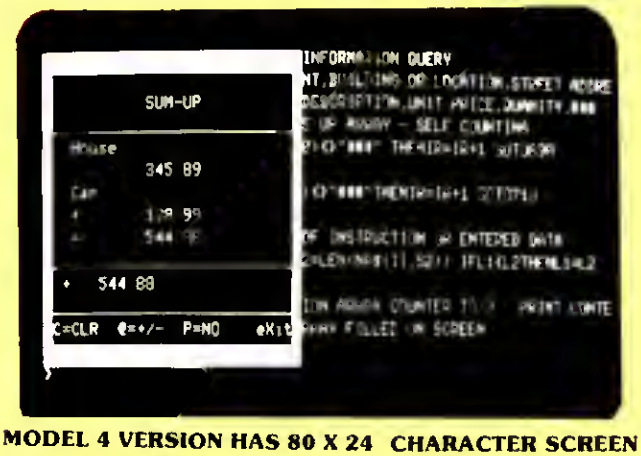
BUY MODEL 4 MULTIDOS
GET ZEUS EDITOR/ASSEMBLER FREE!

Free Zeus comes with reference card of instructions
buy the full manual for $19.95

This offer good until January 1

SUM-UP: The Quick Calculator

never reach for a colculator again!
use SUM-UP instead.



MODEL 4 VERSION HAS 80 X 24 CHARACTER SCREEN

- pops up on the screen, over your application program
- add, subtract, muhiply, divide - algebraic or reverse notation
- works with your word processor, spreadsheet, or any application program
- send output to printer as you enter data
- fluating point math accurate to 8 digits - scientific notation after that
- use text labels
- decimals align
- versions for all major DOS's
- uses only 5.5K of memory

specify Model I, III, or 4

ONLY $24.95

MULTIDOS 80/64 - FOR THE MODEL 4

"...every DOS I've tried has either been too slow and weak or too complicated and powerful for me to use. I feel sure that many other people have this same problem, and MULTIDOS is certainly the solution."
Tim Knight in InfoWorld

MULTIDOS 80/64 runs Model III software, but lets you have Model 4 features...does not run TRSDOS 6 software.

- flip between 64 and 80 characters on the screen; 32 and 40 character widths also available
- runs Model III software
- use your extra 64K memory as a MEMDISK, automatically sets up MEMDISK as system disk, allowing use of the 0 drive for a data disk
- for 4P owners, never load MODEL3/A file again!
- disk I/O code written for Model 4, get fewer errors than you get using a Model III DOS
- a much faster BASIC, many enhancements and debugging tools
- over 41000 free bytes of memory in BASIC
- runs BASIC programs written for the Model III in 64 characters, or easily modified to 80 characters
- write programs using 80 character screen, function keys, and extra memory
- keyboard returns an extended character set, user controllable
- Includes all the new features in 1 7 MULTIDOS

MULTIDOS 80/64 for the Model 4 ..... $99.95
Get MULTIDOS for your Model I or III too...only $89.95

ZEUS EDITOR/ASSEMBLER

- supports undocumented Z80 op codes and standard pseudo-ops
- really FAST assembly
- intelligent error display - shows line number and file containing error, even when you don't print to screen
- easy line editor won't let you enter incorrect syntax
- remembers file name of source and object code; eliminates accidental overwrite
- dynamic renumber; no more "no room between lines"
- calculator mode gives answers in decimal, hex, and binary
- GET command gets files from disk with lightening speed; handles big files so fast you'll think they're small
- doesn't hog memory - lots of room for source code
- easy block move & duplication
- "pages" the screen backward & forward for easy editing
- reads and writes files in ASCII, EDTASM, and Zeus compressed format
- Model 4 version supports 80x24 lines

ZEUS for Model I, III, 4, or MAX-80 ..... $79.95

beot the price rise on Zeus - order now!

AlphaBit Communications, Inc.

☎call (313) 581-2896

13349 Michigan Ave
Dearborn, Michigan 48126

We accept check, credit card, or will send COD Add $2.00 shipping & handling to all orders Add $1 65 for COD Michigan resident add 4% sales tax.

Circle 476 on Reeder Service card.

Leigh L. Klotz of McComb, MS, reports that TRSDOS 6.2 lets you use **periods instead of slashes as separators** when entering the date on boot-up, and notes, "This makes filling in the date from the numeric keypad a snap."

Model 2000 owners: Alice Davis of Columbiaville, MI, sent the short **Basic screen print routine** in Program Listing 3. Line 15 includes the number of lines to print, while line 20 represents the width. You can print portions of the screen by changing the values; for example, to print the lower right quarter, change line 15 to FOR A = 13 TO 24 and line 20 to FOR B = 41 TO 80.

Figure 2 lists **several patches** we've received recently. The first, from Kenneth Stahl of Manassas, VA, prevents Model 4 ALEDIT's J command from erasing the first column of the response. The second two, also from Stahl, let you permanently enable **external drives** 2 and 3, respectively, under TRSDOS 6.2. The next four, 4–7, are from James R. Reed of Dallas, TX. The first eliminates delta symbols used to indicate two spaces in **SuperScript**. The next adds the library command **Kill**, which functions as Remove does. The third disables **password checking**. Finally, the fourth prevents the screen from clearing after a pause when reading long directories.

Patches 8 and 9, from Adam Rubin of Wappingers Falls, NY, disable the carrier detect check so Models fII and 4 **Videotex Plus**, respectively, will run with modems other than the Radio Shack Modem II.■

Program Listing 3. Model 2000 screen print routine.

```
15 FOR A=1 TO 24
20 FOR B=1 TO 80
30 C=SCREEN(A,B)
40 LPRINT CHR$(C);
50 NEXT B
60 NEXT A
```

```
SET *FF TO FORMS/FLT
FILTER *PR TO *FF
FORMS (XLATE=X'401B)
```

Figure 1. JCL file for sending Epson printer codes.

```
1. PATCH ALEDIT/CMD (O05,46=C3 E1 3B:F05,46=C5 D5 E5)

2. PATCH BOOT/SYS.LSIDOS (O02,04=C3:F02,04=C9)

3. PATCH BOOT/SYS.LSIDOS (D02,0E=C3:F02,0E=C9)

4. PATCH SCRIPSIT/CTL (D14,20=10:F14,20=20)
   PATCH SCR35/CTL (D01,30=10:F01,30=20)

5. PATCH SYS1/SYS.LSIDOS (D02,01=4B:F02,01=00)

6. PATCH SYS2/SYS.LSIDOS (D02,33=10:F02,33=20)

7. PATCH SYS6/SYS.LSIDOS (O0A,5F=00 00 00:F0A,5F=3E 69 EF)

8. PATCH VIDTEX/CMD (ADD=73BE,FIND=CB,CHG=C9)

9. PATCH VIDTEX/CMD (X'409A'=0 0 0)
```

Figure 2. Patches.

Circle 440 on Reader Service card.

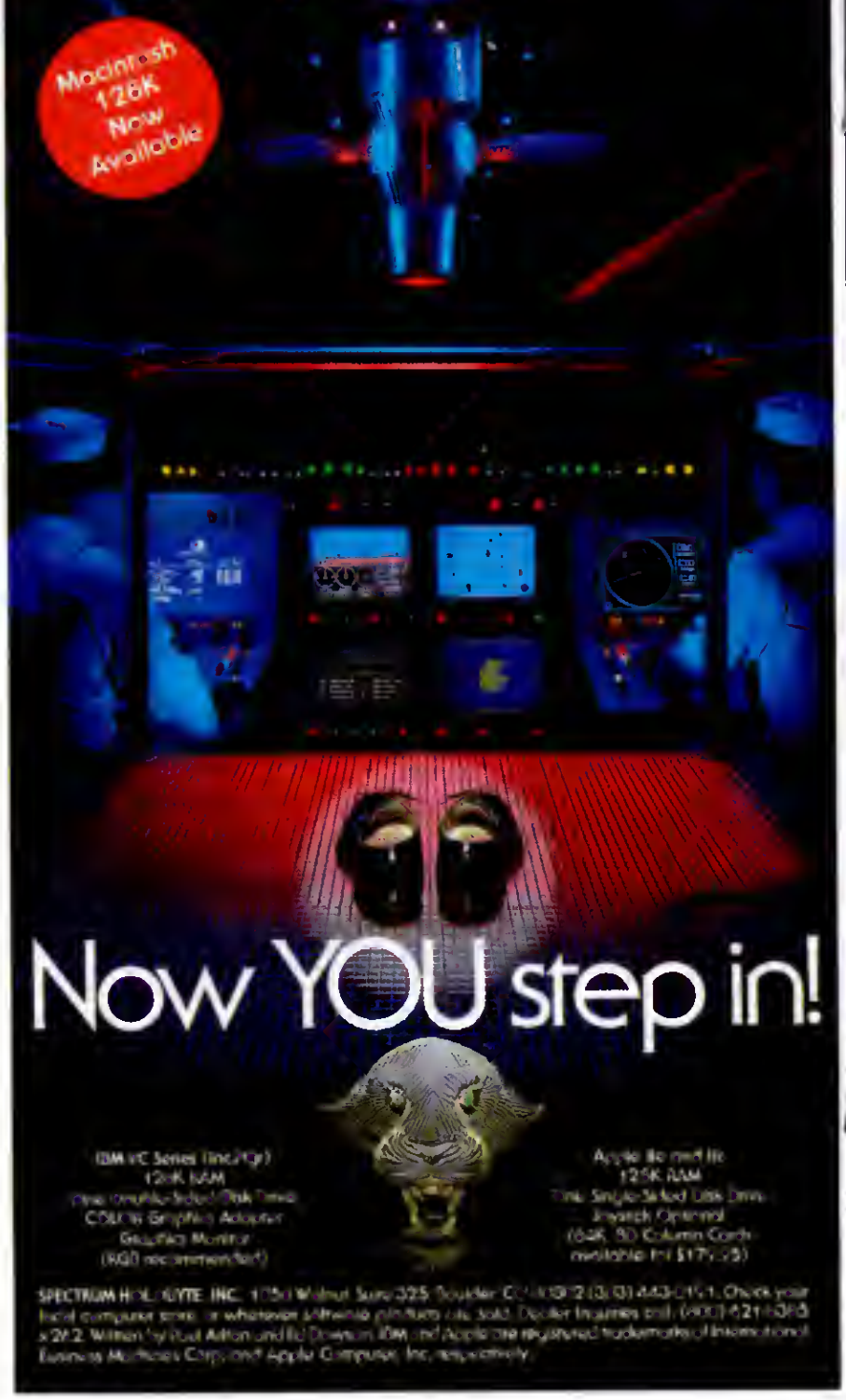
Circle 525 on Reader Service card.

GATO

WW II GATO-Class Submarine Simulation

Spectrum HoloByte Inc.

Macintosh 128K Now Available



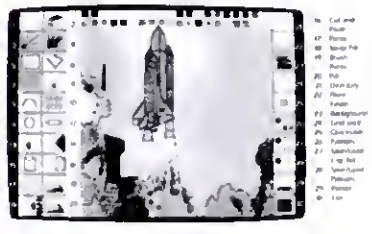
Now YOU step in!

(IBM PC Series Tandy)
128K RAM
One Double-Sided Disk Drive
CGA or Enhanced Graphics Adapter
Graphics Monitor
(RGB recommended)

Apple IIe and IIc
128K RAM
One Single-Sided Disk Drive
Joystick Optional
(64K 80 Column Cards
available for $175.00)

SPECTRUM HOLOBYTE INC. 1050 Walnut Suite 325 Boulder CO (303) 443-0395. Check your local computer store or whatever software outlets are sold. Dealer Inquiries call (800) 821-8385 x262. Written by Bruce Adkin and Ed Dowsett. IBM and Apple are registered trademarks of International Business Machines Corp. and Apple Computer, Inc. respectively.

INTRODUCING...



$49.95

IBM PC, PCjr, XT, AT (and compatibles). 128K RAM. One Double Sided Disk Drive. Color Graphics Adapter. Graphics Monitor. Mouse or Tablet (optional). DOS 2.0 or higher

MAC VERSION NOW AVAILABLE (512K)

TelStar is a leading astronomy program in the K-12 and college science curriculum, lets you create your own planetarium on Apple II and IBM microcomputers. Lesson plans for the use of TelStar including objectives and prerequisites are highlighted in the text "My Students Like Computers" available from Reston Publishing.

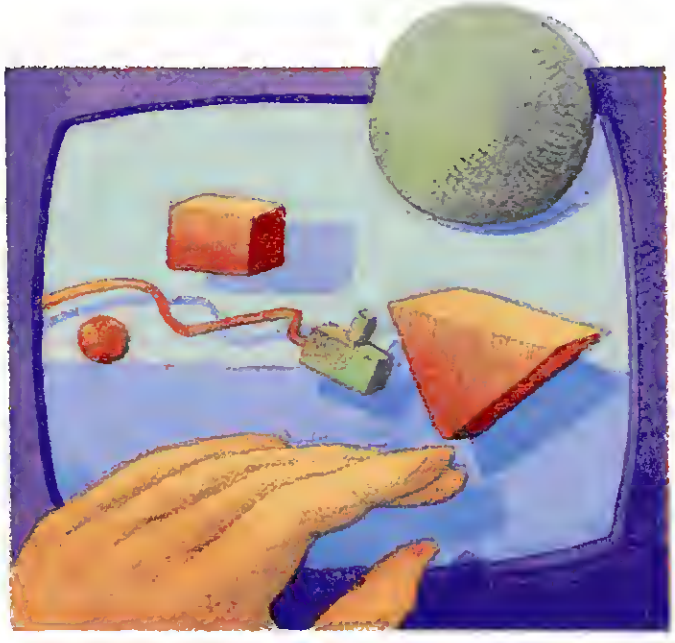
	IBM, XT, JR
*Telstar I One Star Table	$ 49.95
*Telstar II Power Unlocked Star Tables including All Of The Fascinating Messier Objects	79.95
*Telstar III IBM PC's Equipped with An 8087 NDP	129.95

Spectrum HoloByte

TO ORDER CALL
1 (800) 821-8385 X 262

IBM is a registered trademark of International Business Machines Corporation

Top Draw: Micro-Labs' High-Resolution Graphics Tools



by David Engelhardt

★★★★½

GBasic 3.0 runs on the Models III (16K) and 4/4P (64K) and requires a high-resolution board and one disk drive. $49.95 (or free with the purchase of Micro-Labs' high-resolution board). Micro-Labs Inc., 902 Pinecrest, Richardson, TX 75080, 214-235-0915.

Easy to use: ★★★★★
Good docs: ★★★☆☆
Bug free: ★★★★☆
Does the job: ★★★★★

★★★★½

Draw runs on the Models III and 4/4P and requires Micro-Labs' Grafyx Solution or a Radio Shack high-resolution board and GBasic 3.0. Micro-Labs Inc. (see address above). $39.95.

Easy to use: ★★★★★
Good docs: ★★★☆☆
Bug free: ★★★★★
Does the job: ★★★★★

★★★½

The Joy-Mouse Interface works with the Models III and 4/4P and requires a Color Computer joystick, mouse, or touch pad. Micro-Labs Inc. (see address above). $129.95.

Easy to use: ★★★★☆
Good docs: ★★★☆☆
Bug free: ★★★★☆
Does the job: ★★★★☆

Commercial software for the Models III and 4/4P high-resolution (hi-res) boards has been slow in coming, but off-the-shelf products are finally making their way into computer stores. Micro-Labs offers several packages for its hi-res board, including GBasic, Draw, and the Joy-Mouse Interface. The software works on Micro-Lab's Grafyx Solution hi-res board, which is highly compatible with Radio Shack's board.

Overall, I was impressed with Micro-Labs' three products. They represent some of the best graphics products I've seen for the TRS-80-compatible high-resolution boards. While GBasic isn't fully compatible with Radio Shack's BasicG, it has more functions and features. And Radio Shack doesn't have an equivalent to the Draw program or the Joy-Mouse Interface.

GBasic 3.0

GBasic offers more features and versatility than Radio Shack's BasicG, even though it's smaller by about 500 bytes. It also provides wider printer support, including that for Radio Shack, Okidata, Epson, Anadex, Centronics, C. Itoh, and NEC printers.

The GBasic disk contains 40 programs and files of practical applications, demos, examples, and utilities. It requires TRSDOS 6.1.X on the Model 4/4P, with Basic 1.1.0 (other versions of Basic won't work). GBasic will also merge with standard Basic under TRSDOS 1.3, LDOS, DOSPLUS 3.5 and IV (and its extended Basic), and NEWDOS/80. Check with Micro-Labs for the correct version of Basic for proper operation.

GBasic offers some features BasicG doesn't. You can save or load high-resolution screens created with Micro-Labs' Draw program in standard picture file format from within GBasic. You can put an entire screen in reverse video format with a single command. You can even load a version of GBasic into high memory, and call Assembly-language subroutines to perform hi-res functions.

Micro-Labs designed GBasic 3.0 to fully support its hi-res board, but it doesn't completely support Tandy's board. The differences lie mainly in the commands that control screen resolution. Also, Radio Shack's board doesn't allow text overlay of graphics, while Micro-Labs' board does.

GBasic Commands

GBasic links itself to standard Basic: you invoke its functions by preceding commands with the @ symbol. Some of the commands match those of Radio Shack's BasicG, while others differ in both name and operation.

GBasic offers two commands to put you in hi-res mode, one for the Micro-Labs board, the other for the Radio Shack board. Micro-Labs recommends using @ON1 to enable graphics on the Radio Shack board, since it doesn't show hash lines when writing to the display. I found the Micro-Labs board's @ON command much faster in manipulating the display. The @OFF command turns off the hi-res screen and returns you to the normal text screen.

The Micro-Labs hi-res board gives you a choice of display density, which you specify with GBasic's Mode command. In addition to the standard 640- by 240-pixel resolution, you can select resolutions of 512 by 192 pixels and 320 by 240 pixels.

GBasic gives you myriad commands for drawing geometric figures. You set individual points by specifying X,Y screen coordinates and a color parameter that dictates different video densities. Available color values range from zero to 255, which produce "colors" from blanks to solids.

You can test these points to determine their status with the Point command

Illustration by Katherine Mahoney

and a pair of coordinates. The command returns a value of 1 when the point is set, zero if clear, and 2 if it is out of the 640- by 240-pixel graphics boundary.

You draw lines by specifying X,Y coordinates and a color value. Once you draw one line, you can continue to draw others by specifying only endpoints (X2,Y2 coordinates). Each time GBasic draws s line, the previous stop point (X2,Y2) becomes the implied X1,Y1 value for the next line; you just keep supplying X2,Y2 coordinates.

You can also make boxes and circles. The Circle commands not only let you draw circles, they produce ellipses with different aspect ratios, sections of ellipses, and arcs as well.

You can fill in any of GBasic's shapes with the Fill command. You must make sure you enclose the fill area by solid lines or the fill will bleed outside the shape. A Fill parameter lets you stipulate the density of the fill.

You can change every point on the graphics screen to its reverse-video complement with a single command, and you can print text on-screen, specifying where it's to go with X,Y coordinates. You can print text from left to right, sideways from top to bottom, upside down from right to left, and sideways from bottom to top.

GBasic even lets you simulate animation with Get and Put commands. You can put small sections of a display into an array and retrieve it back to the screen in reverse video. You can also And, Or, or XOR the contents of the array to the screen.

You define sections of the screen to be used as plotting areas or windows with GBasic's Using command. You can use the optional Frame parameter to frame the viewing area, fill it in with various patterns, or erase its contents.

A Print command prints your graphics display. An available Printer parameter lets you specify what kind of printer you're using based on a predefined set of printer codes.

Once you finish designing a screen, you can save it to or retrieve it from disk. Since GBasic saves displays in a disk file, you must use standard Basic commands to open and close them. For example, to load in a display file you type in OPEN"R",1, "FILE NAME/XXX":@LOAD:CLOSE.

Utilities

GBasic comes with several utilities. GTest is a small demonstration routine that runs through a series of graphics displays to verify GBasic's operation. While GTest isn't as long or extensive as Radio Shack's BasicG test, it seems to be effective. It also demonstrates some of GBasic's high-resolution displays.

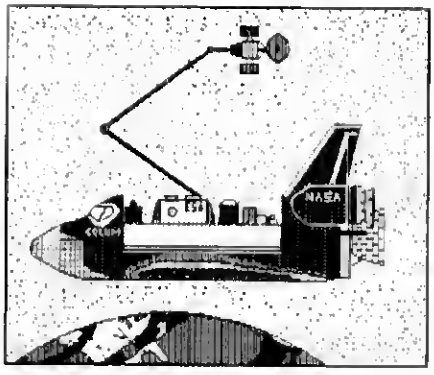


Figure. Printout of a high-resolution design created with Draw.

GBasic/LOD is the Assembly-language program that invokes GBasic's graphics commands. It loads itself into high memory to accommodate machine-language calls to the graphics routines.

SAVLOAD/CMD and SAVLOAD/BAS save and load high-resolution pictures to disk.

Mode V performs the same functions as GBasic's @ON, @ON1, and @OFF commands, and lets you control the hires display from TRSDOS: Setting V equal to zero disables graphics display, to 1 enables the 512 by 192 mode (640 by 240 on the Model 4 board), and to 3 enables 640- by 240-pixel resolution.

The VECTORS/ASM and POINT/ASM source code files contain Assembly-language programs that demonstrate line-drawing, screen-clearing, and point-plotting routines. You'll need an editor/assembler to access them.

The Star Ratings

80 Micro's star ratings reflect our reviewer's impression of a product.

In most cases, the overall rating is an average of the ratings in each of the four specific categories. However, some overall ratings may be higher or lower than this average, depending on the reviewer's subjective opinion.

The stars mean:

★★★★★ Superior;
★★★★ Excellent;
★★★ Good;
★★ Fair;
★ Poor.

The ratings terms translate as follows:

Easy to use: How easy is it for the new user to use the hardware/software/book?

Good docs: Is the documentation clear and helpful in explaining the product's use and anticipating user problems?

Bug free: Did the reviewer encounter any bugs while using the product?

Does the job: How well does the product do what it was designed for?

GBasic's Docs

The GBasic manual is short, but makes up for its brevity with sample demonstration programs written in both Basic and Assembly language. It also provides the high-resolution entry points in upper memory so you can do your own Assembly-language calls to the graphics routines.

Draw

Draw is a 10K Assembly-language program that lets you create and edit sophisticated high-resolution graphics. You use the arrow keys along with Draw's simple commands to create boxes, circles, set or reset points, and so on. You can also shift a screen in any direction, and save any portion of it to disk. In addition, Draw works with Micro-Labs' Joy-Mouse Interface to facilitate drawing.

Draw comes on a TRSDOS 1.3 disk with a few sample high-resolution programs; you have to convert it to use it with TRSDOS 6.X systems. The disk includes a couple of design templates, two "TRON" movie pictures, a dragon, and a picture of the space shuttle.

Draw Commands

Loading Draw and pressing the enter key puts you in Draw's Edit mode. You use the commands displayed on-screen to draw pictures and manipulate the display (Table 1 summarizes Draw's commands).

You draw in one of five modes. The first four, numbered zero to 3, appear on the commsnd list. Mode zero clears every set point the cursor hits. Mode 1 sets every point the cursor hits and mode 2 puts each point in reverse video. Mode 3 lets you skip around the display without affecting the picture and mode 4 lets you enter text on-screen.

Once you position Draw's cursor, pressing the spacebar changes the point under the cursor to reverse video. This lets you do detail work without changing modes. You can clear the display with the clear key, and the break key exits Draw.

Drawing lines and boxes is as simple as positioning the cursor over one point, anchoring it with the E or B key, moving the cursor to the end point (corner point if a box), and pressing the appropriate key again. You draw circles in the same way, except that you have to supply certain parameters to draw ellipses and arcs.

You fill in an area on the display with the F command. It accepts values from zero to 255 so you can stipulate the desired shading or binary bit pattern. Here

again, you have to enclose the area you want to fill to prevent spillover.

You enter text on the screen with the T command, with characters comprising an 8- by 10-dot matrix. It supports upper-/lowercase characters beginning at the current cursor position.

Draw includes two interesting capabilities for screen manipulation. The Negate Screen command puts every point on the screen in reverse video, creating some intriguing results. The Reverse Image command produces a mirror image of the original display. It even displays text in reverse.

You can dump the display to a printer by pressing the D key. The Figure shows a high-resolution space shuttle dumped to my Okidata 92 printer.

Press the S key to save your display to disk in standard SAVLOAD format, which you can load from GBasic or TRSDOS. You can scan any directory by pressing the appropriate drive number key (zero to 3), then decide on which drive to save the display file.

Other Features

You can reposition Draw's entire screen in any direction by pressing the shift and arrow keys. The display moves one dot at a time in the up/down direction and two dots in the left/right direction. Dots that shift off the screen wrap around to the opposite side.

The J command lets you move the cursor directly to a position you specify with X,Y coordinates. Entering X,Y values of zero positions the cursor to the screen's top left-hand corner.

The V command changes the cursor's speed. You can vary the speed in 10 increments, with zero being the fastest. The slowest speed moves the cursor across the screen one dot at a time.

You can change the width of the cursor paint brush by specifying values of from 1–4. This changes the pixel size, which emulates resolutions of 640 by 240 pixels, 320 by 240, 160 by 240, and 160 by 120. You can also create dotted lines when you move the cursor in mode 2 at a "brush" setting of 3 or 4.

The Get and Put commands send and retrieve drawings to and from Draw's 22K memory buffer. You define the portion of the screen you want to save by specifying two opposite diagonal corners of a rectangular area. Then Draw prompts you to name the area with one or two characters. You can save as many blocks as disk space allows or until you fill the 22K buffer.

<ARROW KEYS>—Move cursor	<SPACE>—Complement point
<SHIFT> + <ARROW>—Move screen	<CLEAR>—New picture
<BREAK>—Exit program	<ENTER>—Exit subcommand
0—Clear point mode	1—Set point mode
2—Complement point mode	3—Skip mode
B—Draw a box	C—Draw a circle
D—Dump screen to printer	E—Set line endpoint
F—Fill in shape	G—Get block from screen
J—Jump to position	L—Load hi-res screen
M—Display menu	N—Negate screen
P—Put block onto screen	R—Reverse image L/R
S—Save hi-res screen	T—Text entry mode
V—Velocity of cursor	W—Paint brush width
X—Random X coordinate	Y—Random Y coordinate

Table. Draw commands.

The Put command redraws a previously saved block anywhere on the screen. You position the cursor where you want the upper left corner of the block to begin. When you invoke Put, Draw displays a list of all your saved blocks. You then specify how you want the block put back on-screen; you can copy the block to the screen and overlay the screen's contents, change each point to reverse video, or use the commands And, Or, and XOR to manipulate the screen and block contents.

After putting blocks into Draw's buffer, you can save them to disk for later use. Press the break key, copy down the number that represents the end of the buffer, and exit Draw.

Once in TRSDOS, save your modules to disk with TRSDOS's Dump command: Type in DUMP FILE NAME/CMD (START = X'B9A8', END = X'nnnn').

After saving the modules, type in DO DRAW and answer the prompt to load in LXDraw, which then loads the block module into memory and executes Draw. You can then look at the module names and write them to the screen with the Put command.

Joy-Mouse Interface

The Joy-Mouse Interface, a hardware add-on for the input/output expansion port on the Models III and 4/4P, lets you connect Radio Shack's Color Mouse, Koala Touch Pad, Electronic Book, joystick, or any other Color Computer joystick to your computer. The Interface also provides an on-board expansion connector in case you're using the one on your computer.

The Joy-Mouse Interface works with GBasic, Draw, and Micro-Labs' graphics board, providing direct and quick cursor positioning and drawing. Resolution values of both X and Y coordinates range from zero to 255. The hardware supports both GBasic and Assembly-language programs.

Micro-Labs based the Interface on the ADC0809 8-bit/eight-channel analog-to-digital chip and uses only four of the available eight channels. It offers two joystick modes: proportional and eight-position. The proportional mode, for a Color Computer joystick, varies an analog signal. The eight-position mode works with Atari or Alpha-type joysticks that return a value corresponding to one of eight positions. You need different software for each of the modes.

When using the Joy-Mouse Interface with Draw, you choose from two mouse modes. The first plots a screen resolution of 256 by 240 pixels. The second mode offers full 640- by 240-pixel resolution but divides the screen into three sections with overlaps. (Since the interface can return only X,Y coordinates within the zero to 255 range [due to the 8-bit analog-to-digital converter], it splits the 640 by 240 screen.)

The ? command puts you in the full-screen mode (256 by 240 pixels) and you can set two horizontal dots at a time. The / command puts you in the 640- by 240-pixel mode. Since this resolution splits the screen, the comma key shifts you to the right screen and the period key to the left.

I used the Koala Touch Pad with Draw and found it easy to create drawings. You need to apply constant pressure on the pad while drawing or you'll start splattering dots. While in Draw's Skip mode (mode 3), you draw when you press and hold the left Koala button. For intricate pictures, I recommend using the arrow keys.

Conclusion

If you're interested in high-resolution applications, I think Micro-Labs' software and hardware products offer anything you could want.

I do have one complaint about the manuals' numbering: Micro-Labs skipped some of the numbers and duplicated others. This is a minor point, but Micro-Labs should rectify the problem so that the quality of the manuals matches that of the software and hardware. ■

NEW SORTING UTILITIES FOR TRS-80 MODELS 1 & 3

BSORT51

- Multi-dimension BASIC array sort

BSORT51 is a replacement for the CMD"O" sort of the standard BASIC. Rather than being limited to single dimension string arrays, **BSORT51** can sort one or two dimension arrays of any type - integer, single or double precision, or string. Multiple key arrays may be specified, and the sorting on each key can be done in either ascending or descending order.

Tag arrays (those that do not affect the sort, but merely follow along) may also be specified. **BSORT51** can also create an integer index array without affecting the actual order of the elements in the "sorted" array. For string arrays, "midstring" parameters allow sorting based on a portion or "midstring" of the key array elements.

BSORT51 is entirely machine language, so it is fast. It is invoked off of disk during program execution and will continue with the next statement in the program after execution. This means that NO extra memory is needed to use **BSORT51**.

Order **L-32-200** at **$39** plus S&H.

DSM51

- Disk virtual sorting utility

DSMS1 is THE versatile Disk Sort utility for Model 1 or 3 owners using LDOS 5.1. It is a high speed, disk virtual sorting utility that eliminates the burden of sorting from your applications development project. **DSMS1** will create and maintain index files for you. Since the sort is disk virtual, your only limitation is the amount of available disk space, not available memory!

DSMS1 *can sort random type files consisting of integer,* single and double precision, or ASCII data fields. The file can be up to 65535 records long, with an LRL between 1 and 1024 bytes. Sort fields can be up to 253 characters long. Up to 12 fields can be used as select criteria or sort keys. Any type of relation (e.g. 'equal to', 'less than or equal to', etc.) may be applied to your selection criteria. In addition, logical operators (AND/OR) may be used. For instance: "sort by zip all people with a last name of either Smith or Jones". Any of the 12 specified select fields may also participate in the sort. For example: "sort in zip order and alphabetically by last name within the same zip".

DSMS1 can save a template of the sort/select specifications to a disk file, and may also be run from JCL. This allows even the non-sophisticated user to create index files with a single command.

DSMS1 is 100% machine language, so it is FAST! Compare these sort times to the method you are currently using: Select, Sort, and create an index of 1000 records on two 10 byte ASCII fields, a double precision number, a single precision number, and an integer (34 characters total). With **DSMS1**, the select and sort is done in under 20 seconds from hard disk, and under 40 seconds from floppy.

DSMS1 requires the LDOS 5.1 operating system, and is intended for use with user developed applications or programs that currently use index pointer files. Please note that **DSMS1** creates an index file as opposed to actually re-ordering the data file.

Order **L-35-204** at **$79** plus S&H

Domestic orders add **$1** shipping per product plus **$5** handling for any order not pre-paid by check or money order.

FASTBACK

NEW HARD DISK BACKUP UTILITY FOR TRS-80 MODELS 1, 3 & 4

Replace Tandy's slow file backup program!

Hard disk owners - tired of waiting forever while the HARDCOPY/BAS (Model 1/3) or HDCOPY4/BAS (Model 4) program slowly copies your hard disk file to floppy sector by sector? Do you want to automate your backup procedure and greatly increase its speed at the same time? If so, the **FASTBACK** utility package is for you!

FASTBACK is a 100% machine language program designed to quickly backup e file from hard disk to floppy. It automatically recognizes all floppy formats - single or double sided, 40 or 80 tracks, etc. Built in error checking prevents stopping in the middle of a backup - simply put in a new disk and the process will continue. FASTREAD allows you to restore the file from floppy back to the hard disk should the need occur.

All prompts needed to start these utilities can be answered with a JCL file, allowing you to totally automate the startup. Once started, the programs will prompt for a new disk when needed, and show the next disk number to insert. This makes it possible for an ordinary user to perform perfect backups time after time without fear of mistake.

Typical timings for FASTBACK:

Model 3 Single sided, 40 track	Model 4 Double sided, 40 track
50 seconds, full varify	**95 seconds, full verify**

The **FASTBACK** *package requires* LDOS 5.1 for the Model 1 or 3, and TRSDOS 6.2 for the Model 4/4P.

Order **L-30-055** (Model 1/3) or **L- 30-056** (Model 4), each priced at **$49** plus S&H.

Money Decisions: Bang for the Buck

by Wynne Kelfer

★★★★

The Money Decision Series runs on the Model 4/4P (64K) and requires one disk drive. Tandy/Radio Shack, One Tandy Center, Fort Worth, TX 76102. $49.95 per module.

Easy to use: ★★★★☆
Good docs: ★★★★★
Bug free: ★★★★★
Does the job: ★★★☆☆

The Money Decisions Series is a group of five Model 4 programs that can help you make financial decisions on anything from simple-interest loans to complex real estate investments.

The Programs

Most of us know how to calculate interest earned, but things get more complicated with additions to the initial investment and/or daily compounding. If you throw in tax percentage calculations, you might get lost. That's where the Money Decision Series comes in: It offers virtually any kind of financial analysis you'd want to make. You enter the appropriate data at the prompts, and the program does the hard work.

Tandy sells the series in five independent volumes: Basic Investment Analysis (Money Decisions I), Real Estate and Loans Analysis (II), Business Statistics and Forecasting (III), Business Management (IV), and Advanced Investment Analysis (V). You buy only those programs of interest, and each works similarly; once you use one, you know how to use them all.

Each module displays its available functions on a main menu. After you choose an option, you enter the appropriate variables.

At times, the variables' on-screen descriptions don't clearly indicate what input the program expects—you have to consult the manual. But you can make corrections after you input data, and you can calculate a data value on the fly using the add, subtract, multiply, or divide symbol.

You can display or print out the results of calculations. The on-screen results scroll by, but you stop them by pushing any key. If you print out the results, you can change or reenter the data values to repeat the same equation at the end of the printout.

The Money Decisions modules include on-screen tutorials that describe overall functions and specific sections from within the program. The tutorial moves slowly, which is fine the first time you use it, but it's tiresome if you need information near the end.

You may be able to get along without the manual by using the tutorial if you understand financial concepts. But, if any of them are new, you'll need the manual's detailed explanations.

Strangely enough, you can't save your data to disk, and you lose everything in moving from one module to another.

I did find mention in the Special Options section in Money Decisions IV that you can save your input values and results to a file, but the command doesn't work. And the Special Options table doesn't display this command. It may be that Radio Shack at one time was going to have a Save Files option, but later canceled it and didn't catch this reference.

Simple Investments

The beginning investor or homeowner would probably find greatest utility in the first two modules in the series, Basic Investment Analysis and Real Estate and Loan Analysis. Some of the calculations are quite elementary, so they give the lowest dollar value of the five programs.

Basic Investments helps you calculate personal investment values: how much you must invest periodically to reach a specific goal; how much you can withdraw in equal amounts over a given time span; the interest rate you need to meet a specific goal; the rate of return on investments with differing cash flows; the effects of continuous compounding; the current value of stocks and bonds; and present and future values of annuities. An accompanying chart shows earned interest over a period of time before and after taxes.

The second module, Real Estate and Loan Analysis, lets you figure loan costs from every angle. You can calculate a loan amount from interest, time, and payment amounts; payment amount from time, interest, and principal values; your final payment if you pay off a loan at any point in the payment schedule; time needed to pay a loan at certain interest and payment rates; and interest rates when you have time, payments, and principal data.

You can develop amortization tables for a regular mortgage, as well as calculate adjustable-rate mortgage balloon payments. People planning a mortgage will like the comparison table, which lets you change the loan parameters and see how that affects the terms of the loan. For example, how much more interest will you pay as your mortgage goes from 20 to 25 to 30 years? You can also figure the actual cost of any property, both monthly and total, including the taxes, insurance and utilities.

The second module lets you figure the cost of property, both monthly and total, including taxes, insurance, and utilities.

In typical loans, much of the early payments goes to interest, not principal. This, of course, affects your tax return. Real Estate and Loan Analysis can calculate how much of your payments go to interest, using the Rule of 78s that banks use. It displays the interest for the month, accumulated interest, and interest still owed.

Finally, you can do some rudimentary forecasting, based on past data and smoothed according to your specifications. You can also print out bar graphs of your forecasts, with or without the smoothing constant.

As in all the Money Decisions programs, you can internally pass the results of one computation to another section of the program. I used the program to figure the payment amount for a mortgage, then passed that result to another section to display amortization tables.

Not for the Novice

Money Decisions III and IV, Business Statistics and Forecasting and Business Management, are business financial programs. The first of these is almost entirely devoted to statistical forecasting. You can determine risk-adjusted net present value, expected value of a future event, and average growth rate. You can calculate payoff matrix analyses, Bayesian decision analyses, regression analyses, moving average forecasts, exponential smoothing forecasts, and apportionment by ratios.

A regression analysis, for example, forecasts a future item, such as sales, based on a past correlation between sales and advertising. You may choose linear, geometric, or exponential correlation, but you can't enter more than 24 pairs of values. For each year, you would enter a Y value for sales and an X value for advertising. Unfortunately, you can't label variables in this or any other function. You input and output data in terms of X and Y, and you must remember which is which. At the end, you may enter interpolated X values and see the forecast in Y sales.

The Business Management module includes the following business management functions: lease/purchase analyses; depreciation switches (from accelerated to straight line), rates, and amounts; salvage values; tax depreciation schedules; equipment cost analyses; break-even analyses; linear cost/revenue schedules; fixed and variable production costs; production cost schedules; production alternative cost comparisons and profit/loss; job cost bidding analyses; optimal order and production quantities; inventory reorder and turnover ratios; profit sharing; bonus effects on taxes; and forecasting bar graphs. This program, unlike the others, comes on two disks.

The last program in the series, Advanced Investment Analysis, is strictly for advanced investors. It includes calculations for items like future value (when payments and withdrawals vary), present value of a tax deduction (the deduction being the interest on a loan), current value of a treasury bill (known face value, issue and maturity dates), accrued interest on bonds, and net present value (variable cash flows and periods).

One notable function, called Financial Management Rate of Return, differs from standard internal rate of return calculations by taking into account the cost of financing.

I think Advanced Investment would be highly useful for sophisticated investors. Its functions allow syndicated investment analysis, ratio analysis (of business financial situations), merger evaluation, leverage and earnings per share, and more.

Documentation

Each Money Decisions manual has the same layout. For each function, it explains the calculation, prints the formula, and gives an example. I found the descriptions of the various financial concepts impressive: I understood and used previously unfamiliar ideas.

The docs include a glossary, which defines all the terms, and a special section elaborating on concepts such as compounding, forecasting, and discounted cash flow.

Conclusion

The Money Decisions Series certainly covers the field in terms of financial computations.

However, I was disappointed to find that all the modules cost $49.95. I have no argument with this price for the advanced programs, but this seems steep for the Basic Investment and the Loans and Real Estate packages, which give you fewer useful functions for the money.■

A Disk Zapper With a Difference

by Mark Goodwin

★★☆

Hyperzap runs on the Models I and III (48K) and requires one disk drive. Hypersoft, P.O. Box 51155, Raleigh, NC 27609. $49.95

Easy to use: ★★☆☆☆
Good docs: ★★☆☆☆
Bug free: ★★☆☆☆
Does the job: ★★☆☆☆

Hyperzap is nothing new as a class of software—it's a Model I/III/4 disk zapper—but it does offer some features unique to a utility of this type, including extended directory listings and a memory modification capability. Unfortunately, Hyperzap's inadequate documentation and confusing data entry requirements tarnish its glow.

Hyperzap is versatile: it reads single-, double-, and mixed-density disks. In addition, it automatically detects what brand of double-density board you have and adjusts the disk driver's operation accordingly (since I tested Hyperzap on a Model 4, I was unable to verify this feature).

Features

Hyperzap's main menu presents you with 18 command options (see Fig. 1), many of them standard for a disk zapper: read and write disk sectors, read and format disk tracks, read address marks, position the head to selected tracks, and copy disks.

Hyperzap does offer a unique directory mode, however (see Fig. 2). It displays sequential sector numbers, logical track numbers, spare bytes contained in the address marks, logical sector numbers, sector length codes, data address marks, the memory address for the sectors' data, angular positions, type codes, sector densities, and good or bad CRC values.

While in the directory mode, you can append sector entries; copy the current track entries to the next track; delete, insert, and edit sector entries; generate a standard track; edit sector data; read sectors into memory; change the track bytes; and write sector data to a disk.

Continued on p. 124

```
Screen1:      *** Command Options ***  .  Parameter      Srce  Destn
A Read addr mrks    XC Disk Copy       .  Drive Number   : 00  : 01
C clr Track Table   B AutoBoot Disk    .  No. of tracks  : 40  : 40
D display Table     Z Autopilot        .  Steps/Track    : 01  : 01
I inspect memory    P Change Params    .  Head at track  : 00  : 00
J ## jump @ ##      + Step & repeat    .  Side           : 00  : 00
S Read sectors      T # seek track     .  Size 5/8 inch  : 05  : 05
Q Write sectors     E comb. A,S,D      .  Stepping rate  : 01  : 01
R S/D read track    H(elpful) facts    .  Track offset   : 00  : 00
W Format Track      YX Quit & reboot   .  Sector skew    : 02  : 02

Hyperzap uses    4300-8161      Track/sector table    9000-90A3
Sector data      9C00-9C00      Track buffer          E700-FFFF
Autopilot        9800-9800      P Screen Print          Clear -->
```

Figure 1. Hyperzap's main menu.

```
Screen 2:     #  Tk  Sp  Sc  Ln  CRC  DM  Data  Ang.  TYP  CRC  Den
  Physical  > 01  00  00  00  01  Y    FB  9C00  0673  IBM  Y    S
  track 00    02  00  00  01  01  Y    FB  9D00  4446  IBM  Y    D
   Sector
   Table
  Total 02
  sectors
  Drive 00
  05 inch.

Hyperzap uses    4300-8161      Track/sector table    9000-90B9
Sector data      9C00-9E00      Track buffer          E700-FFFF
Autopilot        9800-9800      P Screen Print          Clear -->
```

Figure 2. Hyperzap's directory mode.

TRS-80™ MODEL 1, 3, AND 4 SOFTWARE

TYPITALL Word Processor $129.95
TYPITALL with Spelling Checker $179.95

Word Processor upwardly compatible with SCRIPSIT — it reads your old SCRIPSIT files and uses the formatting and cursor movement commands you are already familiar with. But it is a completely new word processor with so many advanced features that we can't even mention all of them here.

Send **any** control or graphic/special character to the printer. Control/graphic characters included **in the text** so that you have complete control of all features of your printer. Print the formatted text **on the screen** before going to the printer. Send formatted text to a **disk file** for later printing. **Merge** data from a file during printing. Names, addresses, and other text can be inserted during printing. No need for a separate program for "mail merge" capabilities. Print while editing (spooling). Assign **any sequence of keystrokes** to a **single** control key. Call up to 16 **help screens** at any time. **Move cursor** forwards or backwards by character, word, line, or page. **Reenter** the program with **all text intact** if you accidentally exit without saving the text. Optional spelling checker comes with 29,500 word dictionary. Verify a 3,500 word document in **less than two minutes.** True Model 4 (80 x 24 display, TRSDOS 6) and Model I/III versions.

SYSTEM DIAGNOSTIC $99.95

Is your computer working correctly? **Are you sure?** System Diagnostic has complete tests for every component of your TRS-80 Model 1, 3, or 4 (separate versions necessary for each model).

ROM: checksum test. **RAM:** three tests including every location and data value. **Video display:** character generator, video RAM, video signal. **Keyboard:** every key contact tested. **Line printer:** character tests with adjustable platen length. **Cassette recorder:** read, write, verify data. **Disk drives:** disk controller, drive select, track seek, read sectors, formatting, read/write/verify data with or without erasing, disk drive timer, disk head cleaner. Single or double density, 1-99 tracks. **RS-232-C interface:** connector fault, data transmission, framing, data loop, baud rate generator.

SMART TERMINAL $74.95

The **intelligent** telecommunications program for your TRS-80 Model 1, 3, or 4, or Model 2 CP/M. **Memory buffer** for sending and receiving files. **Automatic transmission** of outgoing data. **Automatic storage** of incoming data. **Character translations.** True BREAK key. Help screens, line feed filters, echo and line printer toggle switches, and more.

TRS-80™ MODEL III ASSEMBLY LANGUAGE $16.95

A complete course in assembly language, written for the **beginner.** Contents include: The Z-80 instruction set, TRS-80™ Model III ROM and RAM; using the Editor/Assembler; reading, printing, and moving data; arithmetic operations with integers; floating-point and BCD numbers; logical and bit operations; cassette input and output; USR subroutines in BASIC; RS-232-C data communications, disk input and output, the TRSDOS 1.3 disk operating system.

MONITOR #5 $22.95
Book and MONITOR #5 $29.95

A comprehensive machine language monitor and debugging program. **Display** memory in ASCII or hexadecimal format. **Disassemble** memory to show machine language commands. **Move and compare** blocks. **Search** and **modify** memory. **Relocate** machine language programs. Read and write **cassette tapes. Unload** programs in low RAM on disk. **Print** optionally on video display or line printer. **Save and load** disk files. **Input and output of disk sectors,** bypassing disk operating system. **Complete debugging package,** including setting and displaying registers, single stepping through machine instructions, setting breakpoints, and executing machine language operations.

TRS-80™ - TANDY - MS-DOS IBM - CP/M SOFTWARE

SMALL BUSINESS ACCOUNTING $99.95

Newly revised, this program is based on the **Dome Bookkeeping Record #612,** and handles **general ledger** and **payroll** for a small business. Category breakdowns are provided for both income and expenses. Monthly, through last month, and year-to-date summaries computed. Start the fiscal year with any month.

Payroll section handles up to 99 employees. Automatic computations for F.I.C.A., federal and state income tax. Three optional deductions also included. Print both payroll and expense checks using same forms. Reports include monthly, quarterly, and year-to-date summaries, 941 and W-2 forms. Simple and easy to learn — ideal for first-time computer users.

HOME BUDGET and CHECKBOOK ANALYST $59.95

A complete checkbook program together with budgeting, income and expense analysis, comparisons, and projections. Enter and print checks, enter deposits, and compute your current checking balance. Program also handles non-check expenses, bank debits, and income. Monthly and year-to-date summaries and yearly projections based on data through a known month. Monthly expenses compared to a pre-established budget.

MAILING LIST $69.95

Build and maintain mailing lists of up to as many names as you can fit on standard diskettes (1,250 for TRSDOS and CP/M, 2,500 for MS-DOS). Four-line labels with optional line that can be used either for unprinted data or as part of the label. Add, change, delete, or find names. Sort according to data in **any** field. Print labels in 1, 2, 3, or 4 adjustable columns.

SMALL BUSINESS MANAGEMENT SYSTEM $299.95

A complete **point-of-sale** program for a small business. Handles **order entry, invoicing, inventory,** and **bookkeeping,** including general ledger, accounts receivable, and accounts payable. Includes up to 999 8-character part numbers. Items deducted from inventory when orders entered. Handles both customer accounts and single orders. Invoices printed on forms or plain paper and include discounts, sales taxes, and shipping and handling charges. General ledger produces monthly and year-to-date totals. Receivables tracked to invoices, automatically updated as income entered. Inventory reports track sales by part numbers.

HOWE SOFTWARE

14 Lexington Road
New City, New York 10956
Information and same day orders
(914) 634 - 1821
24-Hour TOLL FREE Order Number
outside California only
(800) 428 - 7825, ext. 169
Inside California only
(800) 428 - 7824, ext. 169
When ordering, please give your computer model number.
Terms: Checks, Visa, MasterCard or C.O.D.
Shipping and handling, $3.00. Canada, Mexico, Hawaii, $4.00.
Air mail overseas, $12.00. New York residents add Sales tax.
TRS-80 is a trademark of Tandy Corp.

CP/M...The Software Key That Unlocks Your Model 4

CP/M is the standard 8-bit Z-80 operating system and many thousands of programs have been written to run under this system. With Montezuma Micro's CP/M you can run these programs on your Model 4/4P. Think about all those nationally known programs you've wanted to use. Programs like WordStar; dBASE II; SuperCalc; MultiPlan etc. With our version of CP/M 2.2 all those public domain programs on bulletin boards across the USA are available for free downloading. CP/M is the missing link that joins all this software to your Model 4/4P. Montezuma Micro's CP/M comes ready to use and requires no hardware modifications. This product has been awarded the best and highest ratings in the reviews and we are continuously improving it with you in mind. With our CP/M you get more than just a DOS. You get the other half of your Model 4/4P.

AVAILABLE NOW FOR IMMEDIATE SHIPMENT – Less Hard Disk Drive Support..... **$169**
Optional Hard Disk Drive Support ..... $ 30

(Radio Shack 5M, 12M, 15M, 35M • Aerocomp/Percom 5M, 10M, 15M, 30M • Bi-Tech 5M, 10M, 11M, 15M, 20M, 30M, 40M)

FEATURES

- Full range of floppy drive support. Dual-head and/or 80 tracks.
- Optional hard disk support allows positioning and selective assignment of logical drives. Easy backup routine.
- Memory drive allows the use of the other 64K RAM bank on 128K machines.
- Modem 7, a powertul public domain communications program furnished at no charge, allows for file transfer and remote database access such as CompuServ and The Source.
- CONFIG is our flexible utility that allows complete control of all operating parameters from menus. Format, read and write more than 30 different manufacturer's disk formats with more being added rapidly.
- Disk Utility Program allows fast format, backups and verifying of ours and other manufacturer's disks.
- These CP/M utilities are included: ASM; DDT; DUMP; ED; LOAD; MOVECPM; PIP; STAT; SUBMIT; SYSGEN; and XSUB.

WHY BUY OURS?

	MONTEZUMA MICRO	RADIO SHACK
Transient Program Area (TPA)	**55K**	52K
Bytes free in MBASIC	**30,776**	18,488
Bytes free of formatted disk	**196K**	160K
64K Memory drive	**YES**	NO
Double-Side/80tk drive support	**YES**	NO
Format, read/write other CP/M formats	**YES**	NO
Communication program included	**YES**	NO
Share HD with TRSDOS/LDOS	**YES**	NO
Assign multiple drives to HD	**YES**	NO
Boots from Herd Disk (4P)	**YES**	NO
Popular terminal emulation	**YES**	NO
User defined function keys	**9**	3
Timely product support	**YES**	NO

128K MEMORY UPGRADE

Our upgrade kit includes 64K RAM, a geniune PAL chip and instructions for installation. This kit will upgrade your 64K Model 4 to 128K and allow the use of our MEMLINK program and the TRSDOS 6.x MEMDISK. Guaranteed 1 year.

A BARGAIN AT ONLY $74 Model 4
$64 Model 4P – No PAL

This popular software is available for the Model 4/4P using our CP/M.

WordStar Fast memory-mapped version 3.3 ..... $250
MailMerge Multi-purpose file merging program ... 125
SpellStar 20,000 word proof-reader on a stick 125
StarIndex Creates indices and Tables of Content... 85
DataStar Data entry and retrieval is yours..... 175
ReportStar Report generator and file manipulator.... 150
InfoStar The above two programs..... 300
dBASE II with Disk Tutorial..... 385
CBASIC version 2.8..... 85
TURBO PASCAL by Borland. This is the one..... 45

ORDER INFORMATION

Call now and your order will be shipped immediately. We accept American Express, MasterCard and Visa plus we ship COD (cash or cashier's check only). Credit cards are not charged until your order is shipped. Add $4 shipping per item on orders within the 48 states. Suitability of the software is the responsibility of the purchaser as there are **NO REFUNDS.** Defective items will be replaced upon their return, postpaid.

ORDER NOW . . . TOLL FREE
800-527-0347
800-442-1310
The Toll Free lines are for orders only
Specifications subject to change without notice.

CP/M is a Trademark of Digital Research, Inc., Interchange and Memlink are Trademarks of Montezuma Micro. TRS-80 is a Trademark of the Tandy Corporation. WordStar, MailMerge, SpellStar, StarIndex, InfoStar, ReportStar, DataStar, SuperSort and CalcStar are Trademarks of MicroPro International Corporation. Multiplan is a Trademark of Microsoft

MONTEZUMA MICRO

214-339-5104
Redbird Airport, Hangar #18
P.O. Box 32027
Dallas, Tx 75232

"WE KEEP YOU RUNNING"

VISA MasterCard

Born to Run

From its inception, C was developed as an unfettered and transportable language; one C program works unaltered on a number of computer systems. JOHN B. HARRELL III gives you the lowdown and describes its structure and commands.

The babble of languages available for microcomputers makes it hard for a programmer to decide on something new. If you're not satisfied with Basic, Assembly, or Pascal, or if you're curious about other languages, I encourage you to explore C.

I'm not an expert in C, but I've reviewed three exceptional compilers and have gained a real fondness for the language. In this article, I'll introduce some of the concepts that led me to accept C so readily.

A History Lesson

C was developed as a system programming language for the Unix operating system on a PDP-11 minicomputer. The objective was to give the programmer power comparable to Assembly language's without Assembly's tediousness. C was also designed to be portable among a variety of computer systems. The most outstanding example of its power and versatility is Unix itself: Some 90 percent of it is written in C. Unix would not be implemented on so many computers had it been written in another language.

You can best classify C as a medium-level language. Its sophisticated control structures and neat, compact notation are similar to those of PL-1, Pascal, and Algol. However, it lacks many of those languages' features, such as string and data storage manipulation, and advanced input and output facilities.

This weakness is also C's greatest strength. It is relatively uncluttered yet has what you need to manipulate data, much as an assembler does. Thanks to C's minimal structure, a compiler can generate highly efficient code. In fact, compilers on the market today produce better code than most programmers are capable of.

First Words

C is a language of symbols. On first sight, a complex program is enough to make you swear you'll never C. Programs comprise functions, each of which performs a unique task. Each program must have a main function, which is the first part of the code that executes. The standard first example of a C program displays the phrase "hello, world":

```
main( )
{
    printf("hello, world\n");
}
```

The MAIN( ) statement denotes the function the operating system will initialize. The function body starts with a left brace and ends with a right brace. C uses shorthand notation; what could be easier than typing in { and } instead of Begin and End, as you do in Pascal or Algol?

The PRINTF statement is a library function that instructs the computer to display the string on the standard output device. The \n character is C notation for an end-of-line character (other common characters also have special C notations).

Before I move on to a more difficult example, look at Figs. 1–3. Figure 1 lists C's reserved words. Figure 2 lists some common functions a C compiler includes in its standard library. These generally accepted Unix equivalents add all the required functions to the language. Figure 3 describes C's operators—the real power of the language. Using them, you can perform a complex operation in a single statement.

In C, identifiers are composed of letters and numbers. You must make an identifier's first character a letter, however.

C is case-sensitive. For example, identifier "abc" differs from "ABC." You must put all reserved words in lowercase. It's convenient to type in all identifiers and reserved words in lowercase, reserving uppercase for symbolic constants in macros (I'll discuss this later).

Learning to Type

C supports several data types, which generally conform to the basic units of computer physical structure such as bytes, words, or double-words.

The smallest unit of storage is "char," typically a byte long. It can hold one character, and will hold all members of the computer's character set. Characters cannot have a negative value.

The next unit of storage is an "int," or integer value. On a typical microcomputer, this value is a 16-bit word. You can modify an int with "short" or "long" to denote decreased or increased precision (and storage allocation).

You can also designate an integer value as unsigned, which makes the compiler treat the number without regard to sign. For example, a normal integer value on the IBM PC will typically represent values from −32,768 to 32,767. An unsigned integer can assume values of zero to 65,535.

C also supports operations on floating-point numbers such as 6.023×10^{23}. Single-precision numbers are called "float" and double-precision values are called "double." For many microcomputers, float values will have six to seven digits of precision and double values will have about 15 digits of precision.

3-D illustration by Phil Gerace Photograph by Frank Cordelle

The ABCs of Storage

The default storage class is "automatic"; that is, a program automatically allocates variables whenever it executes a function and removes them when the function ends. Automatic variables don't retain their values from one execution of the function to the next.

You can also classify automatic variables as "register" variables, with some restrictions. This tells the compiler to generate code that maintains these values in the computer's registers as long as possible. The program therefore executes faster by using the registers more efficiently.

Sometimes you want variables to retain their last values from one function execution until the next. You do this by declaring the variables as "static"—the compiler will reserve permanent space for them. This might speed up a program by reducing the overhead it takes to allocate and deallocate variables automatically. However, static variables can prevent the code from being reentrant and recursive. You need reentrant code if your program is to be "burned" into a read-only memory (ROM).

Variables can also be "extern," or external, to the function declaring them; the current function block uses them but you define them in some other module. The extern attribute reserves no space in the module where you declare the variable as external.

auto	entry	short
break	extern	sizeof
case	float	static
char	for	struct
continue	goto	switch
default	if	typedef
do	int	union
double	long	unsigned
else	register	while
	return	

Figure 1. C's reserved words.

Name	Description
double atof(cp) int atoi(cp) long atol(cp)	String to double, integer, or long integer conversion.
ftoa(val,buf,prec,type)	Converts from double-precision number to char in a specified format type and precision.
Close(fd) fclose(stream)	Close the file or device pointed to.
open(fd) fopen(stream)	Opens the file or device for input and/or output.
read(fd,buf,bufsize) write(fd,buf,bufsize)	Unbuffered input and output functions.
fread(buf,size,cnt,str) fwrite(buf,size,cnt,str)	Buffered binary file input/output.
fseek(str,offset,origin) lseek(fd,offset,origin)	Reposition a stream or file.
getc(stream) getchar( )	Get next character from an input stream or stdin.
gets(s) fgets(s,stream)	Get a string terminated by a new line character from sdtin or specified stream.
ioctl(fd,cmd,stty)	Set or determine the mode of the console.
char *malloc(size) char *calloc(nelem,size)	Dynamic memory allocation functions.
printf(fmt,[arg]. . .) fprintf(str,fmt,[arg]. . .)	Format print output to stdout or the specified stream.
sprintf(buf,fmt,[arg]. . .)	Format print output to the specified buffer.
putc(c,stream) putchar( )	Put a character to the specified stream or stdout.
puts(str) fputs(str,stream)	Put a character string to stdout or the specified stream.
scanf(fmt[,ptr]. .) fscanf(str,fmt[,ptr]. . .)	Scan stdin input or the specified stream and convert text under format control.
sscanf(buf,fmt[,ptr]. . .)	Scan buffer; convert text under format control.
char *strcat(s1,s2)	Concatenate two strings.
strcmp(s1,s2)	Compare two strings and return result.
char *strcpy(s1,s2)	Copy string s2 to s1.
strlen(s)	Return string length.
char *index(s,c)	Find first occurrence of character in string.
toupper(c) tolower(c)	Converts character c to the designated case.

Figure 2. Partial list of C standard library functions.

Control Structures

The most important control feature in C is the block, a group of statements enclosed in braces { }. These statements (and declarations, too) become one logical statement. I'll use "statement" to mean a single statement or block.

Probably the most common decision statement is If. . .Else, which has the syntax

```
if (expression)
    true-statement;
else
    false-statement;
```

where "else" is optional. If the result of the expression is true (or nonzero), the program executes "true-statement"; otherwise, it executes "false-statement."

Like Pascal, C executes a set of statements until a condition is met in two ways: While and Do. . .while. The difference is that a While statement tests the expression before executing. Do. . .while always executes the statement at least once. Their syntaxes are:

```
while (expression)
    statement;
```

and:

```
do
    statement;
while (expression);
```

A closely related control statement is For, which has the syntax:

```
for (expr1; expr2; expr3)
    statement;
```

The For statement evaluates expr1 as an initializing expression for the loop. Then it evaluates expr2 and tests it. If that value is true, the program executes the statement. It next evaluates expr3 (normally the incremental value for the loop) and repeats the cycle.

C also provides a multipath decision statement, similar to Pascal's Case statement, called Switch, that evaluates an expression and tries to match it to one of

by Daniel Zenzel Jr.

A C interpreter—and seven simple programming examples—get you started with C.

Write Away

My Basic interpreter, C Trainer (see Program Listing 1), will give you an idea of what C is all about without having to buy a C compiler. It's not very powerful, but it will run the C routines I provide. You can also write your own little C programs with it.

You create your C source program in Basic or with a word processor, saving the program in ASCII format. In Basic, you produce left and right braces, respectively, with the clear/shift/< and clear/shift/> keys, and the backslash with the clear/slash combination.

You can include program comments, but be aware that they will strain the capabilities of C Trainer and increase the amount of garbage collection. I find that programs without comments run 20 to 30 percent faster than those with.

Once you save your C program, run C Trainer and enter the name of your source file. After C Trainer loads the program, it automatically forces string garbage collection. If you don't want this, delete line 2480. You'll avoid a delay, but for some programs you'll just postpone it until some time during execution.

Be patient when C Trainer executes a program. The interpreter, since it is in Basic, works slowly. It might even appear at times to hang up. Just give it a little extra time before hitting the break key.

C Trainer only supports the integer type, and not pointers, arrays, or user functions. It can only interpret a MAIN() procedure. I did, however, implement the standard library functions PRINTF, PUTCHAR, and GETCHAR, so that you can have limited input and output from the C program. PRINTF allows the %d options to print integers, and PUTCHAR requires an integer argument. (For PUTCHAR, the argument is the number whose CHR$() you want to print.)

The Figure summarizes the C constructs that C Trainer supports, with their required formats and restrictions. The sample programs in Program Listings 2–8 give examples of the PRINTF and PUTCHAR/GETCHAR functions.

As for arithmetic, I implemented simple expressions only. This means that only simple assignment and addition, subtraction, multiplication, division, incrementation (i + +), and decrementation (i – –) will work. This should be enough to at least get an idea of how C works.

If C Trainer encounters any syntax errors, the interpreter will usually display an error message and stop. This means that all errors in a C program are fatal. At this point you should load your C program back into Basic and correct the error. Some of the error messages aren't the best, but you can easily modify the code to display what you want.

A little tip: When an error stops the interpreter, the variable FPOS contains the relative byte in the source program that was executing when the error occurred. Also, the string array CPROG$() contains the entire C program. You can easily in-

For loops: for (var1 = var2; var1 < = var3; var1 + +)
The comparison must be < =
The initializer must be =
The increment must be + +
Nesting of For loops is not allowed
You can have a While nested in
A single statement or block is OK

While loops: while (var OP var2)
Comparisons OK are <, >, = =, ! =
Var must be variable name
Var2 can be either number or variable
Single statements or blocks are OK
While loops cannot be nested
You can nest a For into a While

If. . .Else: if (var1 OP var2)
Comparisons OK are <, >, = =, ! =
Var1 must be variable name
Var2 can be either number or variable
Single statements or blocks are OK
If statements cannot be nested
You can use For or While in the If

Arithmetic: var = var1 OP var2; var3 + +, var3 – –
OP is +, –, /, *
var1, var2 can be variable or numbers
var, var3 must be variable name

Figure. Supported C constructs.

dex into this array to display the section you had a problem with by using direct Basic commands.

Finally, I've documented the source code, so you can modify it to support different features. ■

You can write to Daniel Zenzel Jr. at P.O. Box 936, Berwick, PA 18603.

LOAD 80

System Requirements

Models 4 and 1000
64 K RAM
Basic

Program Listing 1. C Trainer interpreter.

```
100 ' ---------------------------------------------------------------------
110 ' CTrainer                  Daniel Zenzel, Jr.                  August, 1985
120 '
130 '
140 '   This program will interpret a very small subset of the C Language. The
150 ' input for this program is a C program, created using the standard BASIC
160 ' editor, that was saved with the ASCII option (save "fname",a). This
170 ' interpreter is by no means complete, or for that matter, it does not
180 ' follow the K and R standardization of C.
190 '
200 '   Its purpose is to merely demonstrate the use of the C language as an
210 ' alternative to BASIC, and give one a chance to 'play' with C, in its
220 ' simplest form.
230 ' ---------------------------------------------------------------------
240 DIM CPROG$(1500),FUNCTION.NAME$(5),FUNCTION.LOC%(5), VAR.NAME$(20),
      VAR.INT%(20)
250 CLS:PRINT:PRINT"CTrainer - A 'C'- Language Interpreter  By Daniel Zenzel, Jr
":PRINT
260 INPUT "Enter Source File Name >>";CFNAME$
270 GOSUB 2310
280 PRINT "Interpreting Program...."
290 ' ---------------------------------------------------------------------
300 ' This coda processes global declarations and function declaretiona
310 ' ---------------------------------------------------------------------
320 TOKEN.VAL$="":GOSUB 1820
330 WHILE TOKEN.VAL$<>"MAIN"
340   IF TOKEN.VAL$<>"INT" THEN 420
350      WHILE TOKEN.VAL$<>";"
360        GOSUB 1820: GLOBAL.COUNT = GLOBAL.COUNT + 1
370        VAR.NAME$(GLOBAL.COUNT)=TOKEN.VAL$
380        VAR.INT%(GLOBAL.COUNT) = 0
390        GOSUB 1820
400      NEND
410      GOTO 520
420   IF DELIN$ <> "(" THEN PRINT"Function Declaration Expected":STOP
430   FUNCTION.COUNT = FUNCTION.COUNT + 1
440   FUNCTION.NAME$(FUNCTION.COUNT) = TOKEN.VAL$
450   FUNCTION.LOC%(FUNCTION.COUNT) = FPOS-LEN(TOKEN.VAL$)
460   WHILE TOKEN.VAL$ <> "{": GOSUB 1820: NEND
470   BC%=1
480   WHILE (BC%<>0)
490     GOSUB 1820
500     IF TOKEN.VAL$="{" THEN BC%=BC%+1 ELSE IF TOKEN.VAL$="}" THEN BC%=BC%-1
510   WEND
520   GOSUB 1820
530 NEND
540 ' ---------------------------------------------------------------------
550 ' At this point, we should be at the symbol MAIN(), to start the program
560 ' ---------------------------------------------------------------------
570 GOSUB 1820:GOSUB 1820:GOSUB 1820: ' get to the first statement
580 WHILE (TOKEN.VAL$<>"}")
590   GOSUB 1820:   ' Get statement token
600 IF TOKEN.VAL$="WHILE" THEN GOSUB 3570: GOTO 620
610   IF TOKEN.VAL$="PRINTF" THEN GOSUB 650 ELSE IF TOKEN.VAL$="PUTCHAR" THEN GO
SUB 980 ELSE IF TOKEN.VAL$="FOR" THEN GOSUB 2580 ELSE IF TOKEN.VAL$="IF" THEN GO
SUB 3140 ELSE IF TOKEN.VAL$="INT" THEN GOSUB 1210 ELSE IF TOKEN.TYP=I THEN GOSUB
 1440
620 NEND
630 PRINT:PRINT:PRINT "CTrainer - Done"
640 END
650 ' ---------------------------------------------------------------------
660 ' Routine to handle the printf statement. On entry, fpos will point to
670 ' the left paren of the function call.
680 ' ---------------------------------------------------------------------
690 WHILE CPROG$(FPOS)<>"(" :FPOS=FPOS+1:NEND
700 WHILE (CPROC$(FPOS)<>CHR$(34)): FPOS = FPOS + 1:NEND: FPOS = FPOS + 1
710 B$="": NHILE (CPROG$(FPOS)<>CHR$(34) ): IF CPROC$(FPOS) <> "\" THEN B$ = B$
+ CPROC$(FPOS):FPOS = FPOS + 1: GOTO 740
720 FPOS = FPOS + 1: IF CPROG$(FPOS) = "n" THEN B$ = B$+CHR$(13): FPOS = FPOS +
1 : GOTO 740 ELSE IF CPROC$(FPOS) = "t" THEN B$=B$+CHR$(9): FPOS = FPOS + 1 : GO
TO 740
730 IF CPRGC$(FPOS) = "x" THEN FPOS = FPOS + 3
740 NEND
750 IF CPROC$(FPOS)= CHR$(34) THEN FPOS = FPOS + 1
760 CD%=INSTR(B$,"%d")
770 WHILE CD%<>0 : GOSUB 1820: GOSUB 1820 : ' get comma and identifier
780   IF TOKEN.TYP <>1 THEN PRINT "Printf Syntax Error";STOP
790   CL%=TEMPVAR.COUNT+GLOBAL.COUNT:NHILE VAR.NAME$(CL%)<>TOKEN.VAL$: CL%=CL%-1
: NEND
800   B$ = LEFT$(B$,CD%-1)+STR$(VAR.INT%(CL%))+RIGHT$(B$,LEN(B$)-CD%-1)
810   CD%=INSTR(B$,"%d")
820 NEND
830 GOSUB 1820: ' consume the closing paren
840 PRINT B$;
```

Listing 1 continued on p. 130

the following constant values. If it finds a match, the program executes the statement associated with this constant. The following example demonstrates the Switch statement:

```
switch (input_ch)  {
    case 'A':  statement-1;
               break;
    case 'B':  statement-2;
               break;
    default:   statement-3;
}
```

Switch evaluates the integer expression in parentheses and tries to match it to one of the values indicated in the case labels. If it finds a match, the program continues with the statement associated with that case label. If it doesn't find a match, the statement associated with the default label executes.

The Break statement shunts program execution to the end of the block. Unlike other similar implementations, the switch program flow begins executing on the first match and the program will continue unimpeded to the end of the block. You use the Break statement to force execution of only those statements associated with the selected case label.

While Break forces the program immediately to exit the program control block containing it, this might not be what you want. To skip the remaining statements in the block but continue with the loop until the conditions for termination are satisfied, use the Continue statement.

Since C is a structured language, you can write most programs without GOTO statements, but C's GOTO label statement is there when you need it.

C in Action

Now for some simple programs. My first example uses a standard library function to copy all data from the keyboard to the screen:

```
main( )
{
    int c;
    while ( (c = getchar()) != -1 )
        putchar(c);
}
```

Note the expression in the While statement. The program gets a character, assigns it to the variable c, and tests the result to see if the program detected an end-of-file (–1) indicator. If not, the program sends the character to the standard output device using the PUTCHAR function.

This is an example of the shorthand notation C allows. Why would this program be useful? MS-DOS supports command-line redirection of console input and output from and to other devices or files. If your DOS doesn't support this feature, most run-time packages supplied with commercial C compilers do support it. You could use this simple routine, for example, to copy a file to the video or printer.

Now look at the more complex example in Program Listing 1, Count. This brief

Operator	Description
[ ]	Array subscripting.
–>	Reference to a structure element using a pointer.
.	Reference to a structure element by structure name.
( )	Function calls.
*	Unary * used as a pointer reference.
&	Unary & used as an address reference.
–	Unary negation (two's complement).
!	Unary logical negation (*! expr* yields 1 if expr is false and 0 if true).
~	Unary ~ yields a one's complement of its operand.
+ +	Increment operator. If used before the operand, it is incremented before use; if used after it, it is incremented after use.
– –	Decrement operator. If used before the operand, it is decremented before use; if used after it, it is decremented after use.
(type)	Cast operator. Used to force the conversion of its operand to the specified data type.
sizeof	Returns the size of the operand in bytes.
*	Multiplication: a * b.
/	Division: a / b.
%	Modulus: a % b yields the remainder of dividing b into a.
+	Addition: a + b.
–	Subtraction: a – b.
< <	Left shift: a < < b shifts a left by b bits.
> >	Right shift: a > > b shifts a right by b bits.
<	Tests for a < b and returns truth value.
>	Tests for a > b and returns truth value.
< =	Tests for a < = b and returns truth value.
> =	Tests for a > = b and returns truth value.
= =	Tests for a = b and returns truth value.
! =	Tests for a < > b and returns truth value.
&	Bitwise And operator: a & b.
^	Bitwise Exclusive Or operator: a ^ b.
|	Bitwise Inclusive Or operator: a | b.
&&	Logical And operator: a && b. Left-to-right evaluation is guaranteed and the second operand is not evaluated if the first operand is false.
||	Logical Or operator: a || b. Left-to-right evaluation is guaranteed and the second operand is not evaluated if the first operand is true.
e1?e2:e3	Conditional operator: if expression e1 is true then the result is expression e2 else the result is expression e3.
=	Expression assignment operator: a = b.
+ =	This and the following operators perform assignment of the expression following them to the left-hand value after performing the operation designated. For example: *a op = b* is equivalent to writing the expression as *a = a op b*.
– =	
* =	
/ =	
% =	
< < =	
> > =	
& =	
^ =	
| =	
,	Two or more expressions separated by the comma are evaluated left-to-right and the result of the overall expression is the evaluation of the right-most subexpression.

Note: The operators are grouped in descending order of precedence. Operators have equal precedence within their group.

Figure 3. C's operators.

program will read from the standard input until it detects an end-of-file marker (EOF). As it reads, it counts characters, words, and lines in the text. When it finds the EOF, it displays these totals.

The statements beginning with the # character are called preprocessor statements and direct the compiler to perform specific actions.

The #define statement defines a macro for the compiler that you can use later by referring to that name; in this case, EOF means −1 in the program. These macros can be powerful and can include parameters for substitution into the definition.

The following example of a macro definition produces a function that yields the maximum value of two numbers:

```
#define MAX(A,B) ( (A) > (B) ? (A) : (B) )
```

This expression uses what's called a ternary or conditional operator (expr1 ? expr2 : expr3). It first evaluates expr1; if this expression is true, the result is expr2; otherwise, the result is expr3. I'll return to this later.

Next in Listing 1 comes the header main( ) identifying this as the main program, then declaration of variables. The counters of characters, words, and lines are integers; if you run this on an exceptionally large file (greater than 32K), you should declare them as long integer variables.

The While loop contains the heart of the program. The expression c = getchar( ) reads the next character from the standard input and assigns its value to the variable c. Then, the program checks the character for an EOF. If it finds one, GETCHAR returns a value of −1; otherwise GETCHAR returns the character value. This is the reason for declaring c as an integer value—a char variable is 8 bits and can hold only 256 values, providing no way to distinguish EOF from one of the characters.

When the program reads a character, it increments the character counter [++nc]. When it finds an EOF character, it increments the number of lines [++nl].

Next, the program checks the character for "white space" characters; that is, blanks, tabs, and end-of-line characters (EOLs). The logical operator || (logical or) connects logical tests.

C evaluates expressions containing || from left to right and ends the evaluation when an expression is true. Similarly, the logical operator && (logical and) proceeds from left to right and ends when it evaluates a false expression. This differs from languages such as Pascal or Fortran, which evaluate the entire expression each time it executes before determining its truth value. For example, the Pascal statement:

```
IF X <> 0 AND (1/X) > 3 THEN statement;
```

will always abort on a divide-by-zero error if X is zero. A similar statement using the C operators will not abort.

If the program finds a white space character, it sets the flag variable "inword" to false, indicating that the program is currently not in a word. If it finds another character and inword is false, then the program sets inword true to reflect the start of a word and increments the number of words [++nw].

The last part of the program uses the library routine PRINTF to display its summary. This information outputs to the file "stdout," for which the default device is the system console or video display.

Functioning

The examples I've given so far don't tax the power of C. Now I'll introduce some more advanced features, starting with functions.

In most other languages, functions are separate entities of code that perform some calculations and return a single value. In C, functions describe logical blocks of code that perform a related task. Functions may or may not return a value; they combine the capabilities of Pascal's functions and procedures.

Unlike Pascal, C lets you declare functions in any order within a program module. What's more, you can write and compile functions separately. C encourages you to subdivide your code into logical blocks and to build on these blocks.

Previously, I defined a macro to return the maximum of two numbers (look back at it for a moment). One side-effect of using macros is that the expressions are reevaluated for each repetition of the parameter in the substitution string. In the example above, the compiler evaluates twice the expression you substituted for A and B.

If you need a maximum value function extensively, defining MAX as follows might be much more efficient:

```
int max(a,b)
int a,b;
{
    return ( (a>b) ? a : b );
}
```

This function evaluates only integer parameters, while the macro evaluates a maximum value for any type of data you supply as parameters. You gain efficiency because the compiler generates code to evaluate all parameters prior to calling the function—the function has to work with only a single numerical value for each parameter.

Another benefit of C is its excellent handling of pointers, variables that contain the address of another variable, thereby pointing to the variable. You can use the unary operator * to denote the next operand as the address of a specific type of data item you want to manipulate. The unary operator & instructs the compiler to use the operand's address instead of its value. For example, you could declare ptr as a pointer to a float (float *ptr) and pi as a real variable (float pi), then write:

```
ptr = &pi;
*ptr = 3.14159;
```

The first statement assigns the variable pi's address to the pointer variable, so the second statement is the same as writing pi = 3.14159. If you're confused, my next example should help clarify things.

Arrays are closely related to pointers. In fact, in most cases you can use them interchangeably. Any array operation you can do with subscripting can also be done with pointers.

You define arrays as in most other languages. The statement int numbers[100] defines an array of 100 consecutive integer values that you access via subscript values from zero through 99. Note that the index value begins at zero so the highest

Program Listing 1. Count (from The C Programming Language)

```
/*
   This example program is taken from The C Programming Language by
   Brian W. Kernighan and Dennis M. Ritchie, page 18
*/

#define YES     1
#define NO      0
#define EOF    -1

main()          /* count lines, words, and chars in the input */
{
   int c, nl, nw, nc, inword;

   inword = NO;
   nl = nw = nc = 0;
   while ( (c = getchar() ) != EOF)    {
      ++nc;
      if (c == '\n')
         ++nl;
      if ( (c == ' ') || (c == '\n') || (c == '\t') )
         inword = NO;
      else
         if (inword == NO)    {
            inword = YES;
            ++nw;
         }
   }
   printf("%d %d %d\n", nl, nw, nc);
}
```

End

FASTER THAN TURBO

New lightning-fast ZBASIC zaps the competition.

It's hot. It's brand-new. And light years ahead of anything else.

It's ZBASIC. Written *for* programmers *by* programmers. (If you know BASIC—you know ZBASIC!) Now you can write a program exactly the same way on an Apple, an IBM, a Tandy, or any other major micro and port the source code. You only write the program *once* ...and it runs on all the major micros. The commands stay the same—regardless of the computer* (even graphic commands and disk I/O!).

ZBASIC. Starting now, it's the only language you'll ever have to know.

*subject to hardware limitations.

AVAILABLE NOW!

- Works the same on all micros
- Uses same commands—regardless of computer make
- Structured Basic—(spaghetti optional)
- Device-independent graphics (same graphic commands on all computers)
- 6 – 54 digits of precision (selectable by user)
- Built-in interactive editor and compiler—to compile and execute, just type "run."
- Choice of alphanumerics labels or line numbers
- Chaining with shared variables

Not copy protected—No Run Time Fees or Royalties
One low price gives you everything—there are no hidden costs. Only $89.95 complete.

The finest implementation of the BASIC language for microcomputers!

•=YES N/A=Not available	ZBasic interpreter 3.0	TURBO PASCAL 3.0	MBASIC compiler	True BASIC	BASICA interpreter (IBM PC)
IBM and compatibles	•	•	•	•	•
Apple //e, //c (6502)	•	N/A	N/A	N/A	N/A
Macintosh	4th Qtr	N/A	N/A	N/A	N/A
CP/M 80 2.2, 3.0	•	•	•	N/A	N/A
TRS-80 Model I, III, 4, 4p	•	N/A	•	N/A	N/A
Direct commands	•	N/A	N/A	•	•
Maximum scientific digits of accuracy (COS, SIN, ATN, LOG, EXP etc.)	6 to 54 selectable by the user	11 Binary 8CD N/A	16	16	8
Device Indpdnt Graphics (same CMDS all graphic modes and computers)	•	N/A	N/A	N/A	N/A
SAME File commands all computers?	•	N/A	N/A	N/A	N/A
STRUCTURED Labels, Functions, LONG IF etc	•	•	N/A	•	N/A
Same editor commands all versions/computers	•	•	N/A	N/A	N/A
Sieve benchmark (Byte January 1983, 10 iter's)	13.7 sec	14.1 sec	14.0 sec	261 sec	2190 sec
Shell-Metzner SORT (Sybex-BASIC for Scientist's and Eng. 2,000 5 char. strings)	19 sec.	26 sec	71 sec	194 sec	2700 sec
Executable Machine Lang. & approx. File size	12k	12k	32k	N/A	N/A
PRICE with BCD (BCD=No rounding errors)	89.95	109.95	450.00	N/A	N/A
PRICE without BCD	89.95 BCD FREE	89.95	395.95	149.95	Comes with computer

ZBASIC™

The lightening-fast BASIC from Zedcor

To order use this coupon or call
ORDERS ONLY: 1-800-482-4567
SCHOOLS = Call For Special Package
Utah Residents 1-800-662-5666 Alaska Residents 1-800-982-1500
INQUIRIES : (602) 795-3996

Mail to: ZEDCOR
3438 N. Country Club Road / Tucson, AZ 85716

(Name)

(Address) (Apt.)

(City) (State) (Zip)

(Day Phone)

☐ ZBASIC IBM PC/ Compatible (128K MS DOS 2.0 or better recommended) $89.95 $___

☐ ZBASIC Apple //e, //c (128K Dos 3.3) (Integrated text & graphics) $89.95 $___

☐ ZBASIC CPM 80 (Z 80 2.2 or 3.0) (Provided on 5¼ KAYPRO II-SSDD Format) $89.95 $___

☐ KAYPRO graphics Version (11 & 10) $89.95 $___

☐ ZBASIC TRS-80 (48K) (Circle Model 1,3 or 4 & 2) (Hi Res Bd's Supported) $89.95 $___

☐ ZBASIC Macintosh (Delivery 4th quarter) $89.95 $___

DEVELOPERS – BUY ALL & SAVE $180 $449.75 VALUE $___

Arizona Residents Add 5% Sales Tax $___

SHIPPING U.S. add $5.00 per program $___

Foreign or C.O.D. add $10.00 per program (U.S. Currency only) $___

TOTAL $___

value is one less than the maximum dimension value. C also supports multidimensioned arrays, but you soon learn that you can better write these expressions as arrays of pointers.

I don't want to spend too much time on pointers and arrays, but I'll demonstrate some of their power in a more detailed example (sorry, this one doesn't work with the C interpreter accompanying this article (see p. 41)).

Searching Questions

Program Listing 2, Find, finds a specified string in a text file. It interrogates the command line for parameters and a string to search for. Then it scans input read from the standard input file (stdin), searching for the text string. The optional parameters can specify whether the program displays lines containing the string and whether it displays corresponding line numbers along with the text. The command syntax is FIND [-x] [-n] string, where the x and n parameters are optional and "string" represents any string not containing blanks or other delimiters.

The first statement defines the maximum number of characters you can put on any one line. It uses the preprocessor control statement #define to establish the symbolic name MAXLINE with the proper buffer size.

The main program declaration—main (argc,argv)—tells the compiler that you want to interrogate the command line parameters. The variable argc provides a count of parameters on the command line, including the command name. The variable argv is an array of pointers, each corresponding to the starting character of each command string. Note that you must declare these two variables just after the main program header.

The next statement declares the line buffer and a pointer to a character. The program also declares variables for the line counter and for flags to determine whether to display lines containing the string and their line numbers.

The first While loop scans the command line arguments for the x and n parameters. The first part, --argc > 0, tells the While loop to look at parameters while the parameter count is greater than zero. The -- operator decrements the counter before testing it.

The second part of the While clause tests the first character of the parameter for a leading minus sign, which is required to identify the parameters. The expression (*++argv)[0] == '-' requires detailed explanation. Argv is an array of pointers to character strings. The first pointer is for the command name in some systems. The *++argv says to increment to the next pointer and then use that value. You need parentheses around this expression because of the evaluation priority of the operators * and ++. The [0] looks at the parameter's first character.

and the remaining part of the test compares this parameter to a minus sign.

The program lets you specify the two parameters separately or in one command switch (e.g., -nx). The For statement scans the remaining characters on the selected parameter for valid switch options. The Switch statement checks the options and sets the appropriate flags or displays an error message if the option is invalid.

When the While loop is completed, argc should equal 1, signifying that only the String parameter remains. The If statement checks for a string present and prints an error message and exits if it is not.

The Else clause for this if statement is the heart of the program. It gets a line, checks for the string, performs the designated functions, and continues until there are no more lines in the input stream.

Two functions, Index and GETLINE, do these tasks. The Index function searches the line buffer for the string. If the string is found, Index returns an integer representing the starting position in the buffer. If the string isn't found, Index returns a -1 (this is a normal C function exit showing that the desired function was not done).

GETLINE reads characters from the input file and examines them for an end-of-line character. If it doesn't find an EOL, the program inserts the character into the buffer up to the limit specified. If it finds an EOL, it terminates the buffer as a normal C string (with a zero byte '\0') and returns with the actual length of the line. If no line is available, GETLINE returns a zero value.

Index handles the string and line buffers as character arrays. Note that the function declaration of the two arrays doesn't have to specify the size of the array; it merely tells the compiler that the two variables represent character arrays.

The first For loop initializes the line buffer index "i" and tests the character element s[i] for nonzero. This means the program hasn't reached the end of the buffer. The statement part of this For loop is a block consisting of another For statement and a completion test.

This For loop contains an expression with the comma operator as the initializing expression [j = i, k = 0], which executes once. The loop test consists of two parts: a test to see if t[k] is zero (end of search string) followed by a comparison of the buffer to the string [s[j] = = t[k]].

The last part of the statement consists of two expressions separated by another comma forming the increments for the array indexes. Since the For statement expressions do all the work, no further action is required and the semicolon signifies a null statement.

When the program exits from the For statement, one or both conditions are true: The search string has been exhausted or the string does not match. The If statement tests for a string match and returns an appropriate result.

That's how you do it with arrays. Now

Unions can exist within structures and structures may be in unions. You reference unions using the same operators as for structures.

How Fast Is C?

I included one last programming example as a test of C's performance. Program Listing 3, Sieve, contains source listings in Basic, Pascal, and C for the Sieve of Eratosthenes prime number generator, which has become the *de rigueur* benchmark test. I ran these tests with compilers for Basic, Pascal, and C on a Model 4P and a Tandy 2000. Figure 4 shows the results.

Choosing a Compiler

The compiler is the most important part of any C language software package. Compilers usually generate Assembly-language output that you must assemble. You should get one that generates native Assembly language using standard mnemonics if you want to modify your Assembly code.

For example, Aztec's package generates code in standard assembler format; you can't use it with Microsoft's assembler but Manx's assembler really is better anyway. Your compiler must be able to handle the language as defined in *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie (Prentiss-Hall, $19.95). If you're interested in C, you must have a copy of this book.

Your compiler also must support full preprocessor macro definitions and conditional compilation controls if you're going to easily port your software from one computer to another.

Aztec C prides itself on just this type of support. I have moved programs written for the Model 4 to the Tandy 2000 and IBM PC with relative ease.

Other support software is vital also. This includes the library support. A full Unix-like library is essential. Of the compilers I have seen, Aztec is best in this area. Unix-style utilities are also necessary. Make is a utility that updates complex modular programs by recognizing modules that need recompiling, compiling them, and linking them together. A source level debugging tool helps find those kinky problems that occur from time to time. Another needed tool is a library manager so you can make your own libraries of compiled functions or update existing ones.

On the IBM PC and other 16-bit computers, the compiler should be able to handle all combinations of memory models. This means that you should be able to select code space less than or greater than 64K. Similarly, you should be able to use more than 64K of data space or limit yourself to the smaller configuration. Not all 16-bit compilers support this.

And Finally

C isn't for everyone. It is not a panacea for programming problems. You can do most simple programming tasks in Basic, and C is more difficult to use than many languages. As Fig. 4 shows, compile times are relatively long and can significantly slow program development.

Why, then, is C so popular? It is outstanding for software development. The

Listing 3 continued

```
90 FOR J = 0 TO 8190
100 IF NOT FLAGS(J) THEN GOTO 170
110 PRIME = J + J + 3
120 ' PRINT PRIME,
130 FOR K = J+PRIME TO 8190 STEP PRIME
140 FLAGS(K) = 0
150 NEXT
160 COUNT = COUNT + 1
170 NEXT
180 NEXT
190 PRINT COUNT," primes."
```

(b)

```
program sieve(output);

  const
    size  = 8190;
    sizel = 8191;

  var
    i, prime, k, count, iter : integer;
    flags : array[0..sizel] of boolean;

  begin
    write('10 iterations: ');
    for iter := 1 to 10 do
      begin
        count := 0;
        for i := 0 to size do
          flags[i] := true;
        for i := 0 to size do
          if flags[i] then
            begin
              prime := i + i + 3;
{             write(prime:8);                    }
              k := i + prime;
              while (k <= size) do
                begin
                  flags[k] := false;
                  k := k + prime;
                end;
              count := count + 1;
            end;
      end;
    writeln(count,' primes.');
end.
```

(c)

```
/* Benchmark */
#include <stdio.h>
#define SIZE   8190  /* size of the number array */
#define SIZE1  8191  /* SIZE + 1 */
#define NTIMES 10    /* number of times to execute loop */
#define TRUE   1
#define FALSE  0

char flag[SIZE1];

main() /* compute primes using the Sieve of Eratosthenes */
{
   register int i, j, k, count, prime;
   printf("%d iterations: ",NTIMES);
   for (i = 1; i <= NTIMES; i++)
   {
      count = 0;
      for (j = 0; j <= SIZE; j++)
         flag[j] = TRUE;
      for (j = 0; j <= SIZE; j++)
      {
         if (flag[j])
         {
            prime = j + j + 3;
            /* printf(" %d ",prime); */
            for (k = j+prime; k <= SIZE; k += prime)
               flag[k] = FALSE; /* discard multiples */
            count++;
         }
      }
   }
   printf("%d primes.\n", count);
   exit(0);
}
```

End

Unions can exist within structures and structures may be in unions. You reference unions using the same operators as for structures.

How Fast Is C?

I included one last programming example as a test of C's performance. Program Listing 3, Sieve, contains source listings in Basic, Pascal, and C for the Sieve of Eratosthenes prime number generator, which has become the *de rigueur* benchmark test. I ran these tests with compilers for Basic, Pascal, and C on a Model 4P and a Tandy 2000. Figure 4 shows the reaults.

Choosing a Compiler

The compiler is the most important part of any C language software package. Compilers usually generate Assembly-language output that you must assemble. You should get one that generates native Assembly language using standard mnemonics if you want to modify your Assembly code.

For example, Aztec's package generates code in standard assembler format; you can't use it with Microsoft's assembler but Manx's assembler really is better anyway. Your compiler must be able to handle the language as defined in *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie (Prentiss-Hall, $19.95). If you're interested in C, you must have a copy of this book.

Your compiler also must support full preprocessor macro definitiona and conditional compilation controls if you're going to easily port your software from one computer to another.

Aztec C prides itself on just this type of support. I have moved programs written for the Model 4 to the Tandy 2000 and IBM PC with relative ease.

Other support software is vital also. This includes the library support. A full Unix-like library is essential. Of the compilers I have seen, Aztec is best in this area. Unix-style utilities are also necessary. Make is a utility that updates complex modular programs by recognizing modules that need recompiling, compiling them, and linking them together. A source level debugging tool helps find those kinky problems that occur from time to time. Another needed tool is a library manager so you can make your own libraries of compiled functions or update existing ones.

On the IBM PC and other 16-bit computers, the compiler should be able to handle all combinations of memory models. This means that you should be able to select code space less than or greater than 64K. Similarly, you should be able to use more than 64K of data space or limit yourself to the smaller configuration. Not all 16-bit compilers support this.

And Finally

C isn't for everyone. It is not a panacea for programming problems. You can do most simple programming tasks in Basic, and C is more difficult to use than many languages. As Fig. 4 shows, compile times are relatively long and can significantly slow program development.

Why, then, is C so popular? It is outstanding for software development. The

Listing 3 continued

```
90 FOR J = 0 TO 8190
100 IF NOT FLAGS(J) THEN GOTO 170
110 PRIME = J + J + 3
120 ' PRINT PRIME,
130 FOR K = J+PRIME TO 8190 STEP PRIME
140 FLAGS(K) = 0
150 NEXT
160 COUNT = COUNT + 1
170 NEXT
180 NEXT
190 PRINT COUNT;" primes."
```

(b)

```
program sieve(output);

  const
    size  = 8190;
    size1 = 8191;

  var
    i, prime, k, count, iter : integer;
    flags : array[0..size1] of boolean;

  begin
    write('10 iterations: ');
    for iter := 1 to 10 do
      begin
        count := 0;
        for i := 0 to size do
          flags[i] := true;
        for i := 0 to size do
          if flags[i] then
            begin
              prime := i + i + 3;
              write(prime:0);
              k := i + prime;
              while (k <= size) do
                begin
                  flags[k] := false;
                  k := k + prime;
                end;
              count := count + 1;
            end;
      end;
    writeln(count,' primes.');
end.
```

(c)

```
/* Benchmark */
#include <stdio.h>
#define SIZE   8190  /* size of the number array */
#define SIZE1  8191  /* SIZE + 1 */
#define NTIMES 10    /* number of times to execute loop */
#define TRUE   1
#define FALSE  0

char flag[SIZE1];

main() /* compute primes using the Sieve of Eratosthenes */
{
   register int i, j, k, count, prime;
   printf("%d iterations: ",NTIMES);
   for (i = 1; i <= NTIMES; i++)
   {
      count = 0;
      for (j = 0; j <= SIZE; j++)
         flag[j] = TRUE;
      for (j = 0; j <= SIZE; j++)
      {
         if (flag[j])
         {
            prime = j + j + 3;
            /* printf(" %d ",prime); */
            for (k = j+prime; k <= SIZE; k += prime)
               flag[k] = FALSE; /* discard multiples */
            count++;
         }
      }
   }
   printf("%d primes.\n", count);
   exit(0);
}
```

End

	C	Pascal	Basic Interpreter	Basic Compiler
Model 4/4P				
Source file size	836	811	344	344
Execution file size	8,785	19,076	21,927	33,092
Source time (sec)	123	62	N/A	179
Execution time (sec)	27.1	175	945	20.3
Tandy 2000				
Source file size	896	896	384	384
Execution file size	3,942	27,148	52,672	23,248
Source time (sec)	62	84	N/A	67
Execution time (sec)	3.3	4.2	569	6.0

The code size listed for the Basic interpreter includes the size of the interpreter itself. The compilation times listed include the time required to assemble, link, and/or convert the source code into a stand-alone program.

Model-4/4P: TRSDOS 06.02.00 Disk Operating System
Microsoft BASIC Interpreter 01.01.00
Microsoft BASCOM Compiler version 5.35
Manx Aztec-C80 Version 1.06B
TRS-80 (Alcor) Pascal 02.00.00.

Tandy 2000: MS-DOS Disk Operating System Version 02.11.02
Microsoft BASIC Interpreter 01.03 00
Microsoft BASCOM Compiler Version 5.50
Manx Aztec-C86c Version 3.20C
Microsoft PASCAL Version 3.13.

Figure 4. Sieve of Erotosthenes comparison.

biggest cost factor in developing software is the time required to design, write, and debug the code. If you can reduce any of these factors, your profit will increase. C does this in a major way, since it makes coding routines in Assembly language (a lengthy process) virtually unnecessary.

Other important factors in software design are overall size and speed. As you can see from the simple example in Fig. 4, the code a good C compiler produces is far smaller than that of other compilers for microcomputers. Also, C's compiled code executes as fast as, and often faster than, that of other compilers.

If you're interested in programming applications software and want to exploit your computer fully, you must have a compiler. C lets you run your software on the widest possible variety of systems, and I highly recommend that you investigate it. ■

John B. Harrell III writes about programming and edits Spreadsheet Beat. You can contact him c/o 80 Micro, 80 Pine St., Peterborough, NH 03458.

Powerful Programming Tools At Bargain Prices

C compiler

for the model 1 or 3 using TRSDOS, LDOS, NEWDOS, DOSPLUS, or MULTIDOS; includes full screen text editor and advanced development package

List Price ~~$250.00~~
Sale Price $89.95

This is a full K & R standard implementation of C that includes a Unix compatible function library. The package also includes a 450 page manual with a tutorial on using the C language. If you've been wanting to learn C, this is the package you need.

Features Include

char	8 bits	initializers
short	8 bits	typedef
int	16 bits	static
unsigned	16 bits	auto
long	32 bits	extern
float	32 bits	struct/bit fields
double	64 bits	union

Execution speed on the Model 3 for 10 iterations of the prime number program published in Byte, Jan 83, page 284.

LC Compiler	105 secs.
Alcor C	78 secs.

Special Bonus

Buy one version for $89.95 and get the version for the other model for only $21.

Multi-Basic compiler

for the model 1 or 3, or 4 using TRSDOS, LDOS, NEWDOS, DOSPLUS, or MULTIDOS; includes full screen text editor and advanced development package

List Price ~~$250.00~~
Sale Price $89.95

Multi-Basic is a TRS-80 BASIC compatible compiler. The Model 4 version supports everything in the TRSDOS 6 BASIC interpreter except the COMMON statement. The same support is provided in the Model 1 and 3 versions so programs are portable. The CMD statement is the only statement from the Model 1 and 3 BASIC interpreters that is not supported.

Multi-Basic also supports advanced language features like multi-line procedures and functions, recursion, and dynamic string management (no long pauses for garbage collection).

Execution speed on the model 3 for 10 iterations of the prime number program published in Byte, Jan 83, page 286.

BASIC Interpreter	4570 secs.
Multi-Basic	89 secs.

Special Bonus

Buy one version for $89.95 and get versions for the other two models for only $21 each.

Sale Price Extended Through October 31

C Compiler

Circle version(s)	Model 1	Model 3
One version ($89.95)		
Both versions ($110.95)		
Add 6% sales tax (Texas only)		
Shipping $6 USA/$28 foreign)		
Total		

Name _____
Street _____
City _____
State _____
Zip _____
Country _____
Phone _____

Also available for CP/M & MSDOS $89.95 each

MC ☐ Visa ☐ Money Order ☐ Check ☐ COD ☐

Card # _____ exp _____

Multi-Basic Compiler

Circle version(s)	Model 1	Model 3	Model 4
One version ($89.95)			
Two versions ($110.95)			
Three versions ($131.95)			
Add 6% sales tax (Texas only)			
Shipping $6 USA/$28 foreign)			
Total			

ALCOR Systems
1132 Commerce
Richardson, TX 75081
(214) 238-8554

Multi-Basic is a trademark of Alcor Systems
TRS-80 is a registered trademark of Tandy Corp.
MSDOS is a trademark of MicroSoft
CP/M is a trademark of Digital Research
Unix is a trademark of Bell Laboratories
LC is a trademark of Misosys

Circle 215 on Reader Service card.

NET RESULTS

by David H. Pleacher

Hoops covers the court in recording and reporting basketball statistics by team or player.

Reading through a sheet of basketball statistics may not substitute for the give and take of live play but, as any high school coach can tell you, the numbers give you the lowdown on team performance. My Model I/III/4 basketball statistics program, Hoops, keeps track of a team's record, an individual's record, and overall game statistics (see the Program Listing and Fig. 1).

Hoops lets you print out five different reports: the team record (see Fig. 2), team totals by game (see Fig. 3), cumulative totals for each of the team members (see Fig. 4), an individual player's statistics, and the printout for a particular game.

Getting Organized

The key to Hoops' statistical manipulation lies in its file handling (see the Table for Hoops' line descriptions). The program uses both random-access and sequential files; it opens random-access files using Basic's buffer 1 and sequential files with buffer 3.

Hoops records up to 20 players' statistics in random-access files called PLAYER1/TXT, PLAYER2/TXT, and so on. Each of these files contains records; record 1 holds the statistics for game 1, record 2 for game 2, and so on. Hoops stores the team totals for each game in the random-access file called PLAYER21/TXT, and the opponents' totals for each game in PLAYER22/TXT.

Hoops also uses five sequential files. TEAMINFO/TXT contains the school's (or team's) name, the coach's name, the year, the number of wins and losses, and the number of players on the team. Games/TXT contains the name of the opponent, the date, whether it's a home or away game, and the score for each game.

Players/TXT contains the names and jersey numbers of all the players. Hoops keeps the cumulative totals for a team in Totals/TXT. It initially fills this file with zeros. Hoops uses one other file, TEMPFILE/TXT, when you make corrections; the program opens it through buffer 2.

File-Handling Routines

To see how Hoops' file-handling routines work, follow the routine for adding a player to the team (lines 4470–4570). When you select the option to add a player from the main menu, Hoops first opens the sequential file TEAMINFO/TXT for input and reads the data from it. Then it opens the sequential file Players/TXT for input and reads the data from that file. Finally, it opens the sequential file Totals/TXT and reads the totals for each player from that file.

After you enter the additional players and their corresponding jersey numbers, Hoops opens the sequential files Players/TXT and Totals/TXT for output, and writes the updated data to them. For example, if you just added the 16th player to a team, the routine opens the random-access file PLAYER16/TXT and fields it. This file contains player 16's statistics for each game. If you already played four games when you add player 16 to the team, Hoops fills the first four records with zeros using the RSET (lines 790–860) and Put (line 880) statements.

Often, Hoops accesses several files to make one printout. For example, to print the statistics for the third game, you need the sequential files TEAMINFO/TXT, Games/TXT, and Players/TXT; and record 3 of each of the random-access files PLAYER1/TXT, PLAYER2/TXT. . .PLAYER22/TXT.

Using Hoops

Use the template in Fig. 5 to record game statistics during play. You can later add this data to the program's statistical files.

To use Hoops, enter Basic with three variable files and run the program. (Hoops has a fun but time-consuming opening display. Delete lines 80 and 5150–5350 to eliminate it.)

To enter data for a new team, select option A from the main menu. Hoops prompts you for the school (or team) name, the coach's name, and the players' names and jersey numbers. If you make an error, Hoops lets you correct it at the end of each record input.

Once you enter this information, you can choose any of Hoops' other options: add a player, type in statistics for a game, make corrections to previously entered data, or print out reports. If, by mistake, you select the option to update statistics or the option to add a player, you are given a chance to exit from that module immediately.

Hoops lets you enter a zero for a particular category by pressing the enter key. This is useful when a team member plays only two minutes in a game and most of that player's statistics are zeros.

To print out the statistics, you need a printer with a 110-column capability. You must use 11- by 14-inch paper if your printer prints 132 columns, condensed printing if you have an 80-column dot-matrix printer, or elite printing for a daisy-wheel printer. Feeding single sheets of 8½- by 11-inch paper sideways makes excellent printouts on a daisy-wheel printer.

Hoops' Limitations

You can enter only the 13 statistics the program uses. Although these are probably the most basic basketball statistics, some coaches might keep additional statistics, like minutes played.

You need to keep players' names to 20 characters, and opponents' names to 14. I did this to keep the printouts to 110 columns.■

Write to David H. Pleacher at 5047 Caroline Ave., Stephens City, VA 22655.

LOAD 80

System Requirements

Models I and 4 with changes
Model III
32K RAM
Disk Basic
Printer (110-column)

Illustration by Earl Kelemy

Model I change:

Remove POKEs.

Model 4 changes:

Correct PRINT @ locations.
Change 960 to 1200 in lines 160 and 200.
Remove POKEs.
Delete opening display: lines 80 and 5150–5350.
Change % to \ (clear key with ? key).

Figure 1. Program changes for the Models I and 4.

John Hendley High School Basketball Statistics
1984 - 85

Coach: Bill Isherwood Won: 9 Lost: 1

Game	Date	Opponent	Score Us	Score Opponent
1	12/11/84	Clarke County	69	63
2	12/14/84	Warren County	64	57
3	12/18/84	Parkview	61	51
4	12/20/84	Martinsburg	60	62
5	12/27/84	Harrisonburg	53	59
6	12/28/84	James Wood	82	48
7	01/04/85	Broad Run	68	60
8	01/11/85	Loudoun County	67	56
9	01/12/85	Loudoun Valley	63	62
10	01/19/85	Osbourn	84	68

Figure 2. The team record.

John Hendley High School Basketball Statistics
1984 - 85

Coach: Bill Isherwood Won: 9 Lost: 1

G#	Date	Opponent	H/A	FGM	FGA	FG %	FTM	FTA	FT %	Pts.	Sco. Avg.	Off Reb	Def Reb	Tot Reb	Reb. Avg.	Trn Ovr	A	Drw Fou	Stl	Blk Sht	F
1	12/11/84	Clarke County	Away	32	70	45.7	5	16	31.2	69	69.0	23	22	45	45.0	18	16	1	9	6	18
2	12/14/84	Warren County	Home	26	65	40.0	12	29	41.3	64	64.0	19	20	39	39.0	17	15	0	20	7	18
3	12/18/84	Parkview	Away	25	50	50.0	11	16	68.8	61	61.0	11	19	30	30.0	9	19	1	7	4	13
4	12/20/84	Martinsburg	Home	25	50	50.0	10	28	64.3	60	60.0	12	15	27	27.0	17	13	1	10	4	16
5	12/27/84	Harrisonburg	Home	23	63	36.5	7	12	58.3	53	53.0	17	16	33	33.0	14	14	0	13	6	17
6	12/28/84	James Wood	Home	33	65	50.8	16	21	76.2	82	82.0	16	17	33	33.0	17	14	0	21	6	15
7	01/04/85	Broad Run	Away	25	49	51.0	18	25	72.0	68	68.0	7	15	22	22.0	19	15	1	17	2	15
8	01/11/85	Loudoun County	Home	28	57	49.1	11	18	61.1	67	67.0	17	17	34	34.0	21	17	0	14	4	17
9	01/12/85	Loudoun Valley	Away	24	59	40.7	15	23	65.2	63	63.0	17	16	33	33.0	11	15	0	10	2	20
10	01/19/85	Osbourn	Home	31	58	53.4	22	29	75.9	84	84.0	7	16	23	23.0	12	18	1	16	2	15
		Totals		272	586	46.7	135	217	61.4	679	67.9	146	173	319	31.9	155	156	5	137	43	164

Figure 3. The team totals by game.

John Hendley High School Basketball Statistics
1984 - 85

Coach: Bill Isherwood Won: 9 Lost: 1

10 Game Totals

#	Player	G	Qtr	FGM	FGA	FG %	FTM	FTA	FT %	Pts.	Sco. Avg.	Off Reb	Def Reb	Tot Reb	Reb. Avg.	Trn Ovr	A	Drw Fou	Stl	Blk Sht	F
10	Dwayne Richardson	10	21	1	9	11.1	2	7	28.6	4	0.4	1	2	3	0.3	0	8	0	7	0	6
12	Evan Humbert	10	37	19	45	42.2	25	34	73.5	63	6.3	7	23	30	3.0	25	65	2	20	1	25
13	Herold Brown	3	4	2	9	22.2	0	1	0.0	4	1.3	0	1	1	0.3	0	0	0	0	0	0
14	Jason Morgan	10	38	73	140	52.1	25	43	58.1	171	17.1	27	16	43	4.3	31	36	1	40	3	31
20	Joe Wilson	9	34	24	50	48.0	14	23	60.9	62	6.9	21	19	40	4.4	17	12	2	28	5	27
32	John Morgan	10	38	78	160	48.8	42	63	66.7	198	19.8	39	43	82	8.2	27	16	0	15	24	23
54	Mike Hardware	10	36	36	75	48.0	5	12	41.7	77	7.7	29	37	66	6.6	18	6	0	10	8	22
44	Mario Pritchett	10	33	16	41	39.0	12	17	70.6	44	4.4	5	11	16	1.6	12	14	0	10	0	9
40	Richard Pell	10	30	12	34	35.3	6	8	75.0	30	3.0	0	12	20	2.0	10	1	0	2	1	14
34	Mike Look	8	18	4	12	33.3	4	7	57.1	12	1.5	1	4	5	0.6	3	8	0	3	0	2
21	Jeff Veel	3	4	3	5	60.0	0	1	0.0	6	2.0	3	3	6	2.0	3	1	0	0	0	1
22	Daniel Robinson	4	5	2	3	66.7	0	0	0.0	4	1.0	2	0	2	0.5	1	1	0	1	0	2
30	Jude Sparrow	6	8	2	3	66.7	0	1	0.0	4	0.7	3	2	5	0.8	0	0	0	1	1	2
-	Team Totals	10	40	272	586	46.4	135	217	62.2	679	67.9	146	173	319	31.9	155	156	5	137	43	164
-	Team - per game	1	4	27	59	46.4	14	22	62.2	68	67.9	15	17	32	31.9	16	16	1	14	4	16
-	Opponents	10	40	240	535	44.9	106	162	65.4	586	58.6	153	167	320	32.0	0	0	0	0	0	155
-	Opp. - per game	1	4	24	54	44.9	11	16	65.4	59	58.6	15	17	32	32.0	0	0	0	0	0	16

Figure 4. The cumulative totals for team members.

John Handley High School Basketball Statistics
1984-85
Game #6 Opponent: James Wood Home

#	Player	Qtr	FGM	FGA	FG%	FTM	FTA	FT%	Pts	Scg Avg.	Off Reb	Def Reb	Tot Reb	Reb Avg.	Trn Ovr	A	Drw Fou	Stl	Blk Sht	F
10	Dwayne Richardson	3	0	2	0.0	2	2	100.0	2	2.0	0	1	1	1.0	4	1	0	3	0	2
12	Evan Humbert	3	2	3	66.7	8	9	88.9	12	12.0	1	0	1	1.0	1	3	0	2	0	3
13	Harold Brown	2	2	7	28.6	0	0	0.0	4	4.0	0	1	1	1.0	0	0	0	0	0	0
14	Jason Morgan	3	7	11	63.6	3	3	100.0	17	17.0	2	2	4	4.0	1	4	0	2	0	2
20	Joe Wilson	4	2	6	33.3	0	0	0.0	4	4.0	3	1	4	4.0	0	3	0	6	1	1
32	Joh																	3	2	

Figure 5. Template for recording game statistics.

Legend

Qtr	Quarter	FTM	Free throws made	Def Reb	Defensive rebounds	Drw Fou	Draw offensive fouls
FGM	Field goals made	FTA	Free throws attempted	Tot Reb	Total rebounds	Stl	Steals
FGA	Field goals attempted	FT%	Free throws percentage	Trn Ovr	Turnovers	Blk Sht	Blocked shots
FG%	Field goals percentage	Dff Reb	Offensive rebounds	A	Assists	F	Fouls

Lines	Description
70–130	Main program.
150–340	Most-often-used subroutines.
350–1770	Frequently used subroutines.
1780–1970	Main menu.
1980–2490	Statistics update module.
2500–2580	"Team record" module.
2590–2710	"Team totals by game" module.
2720–2890	"Cumulative totals for team members" module.
2900–3100	"Statistics for individual player" module.
3110–3370	"Statistics for a particular game" module.
3380–4460	Change statistics—make corrections.
4470–4570	"Add team members" module.
4580–4800	Initialization routine.
4810–5110	Program instructions.
5120–5140	Housekeeping.
5150–5350	Opening display.
5360–5400	Error-handling routines.

Table. Program outline of Hoops.

Program Listing. Hoops.

```
10 REM  ** Basketball Statistics **
20 REM  ** David Plescher **
30 REM  ** John Handley High School **
40 REM  ** P.O. Box 910, Winchester, VA 22601 **
50 REM
60 REM  **  Main Program  **
70 CLEAR 500:ON ERROR GOTO 5370
80 GOSUB 5160        ' Opening Display
90 GOSUB 5130        ' Housekeeping
100 GOSUB 4820       ' Instructions
110 GOSUB 1790       ' Menu
120 GOSUB 1590       ' End of Program
130 END
140 :
150 REM  ** Subroutine to press <ENTER> to continue  **
160 PRINT@960, "Press <ENTER> to continue.";
170 IF INKEY$ <> CHR$(13) THEN 170
180 CLS : RETURN
190 REM  ** Subroutine for  YES/NO answer  **
200 PRINT@960, "Is this information correct (Y/N) ?";
210 POKE 16409,1:T$=INKEY$
220 IF T$ <> "Y" AND T$ <> "N"  THEN 210
230 POKE 16409,0 : RETURN
240 REM  ** Print to TEAMINFO/TXT  file  **
250 OPEN"O",3,"TEAMINFO/TXT":PRINT#3,S$;",";C$;",";Y$;",";G;W;L;P:CLOSE:RETURN
260 REM  ** Input from TEAMINFO/TXT  file  **
270 OPEN"I",3,"TEAMINFO/TXT":INPUT#3,S$,C$,Y$,G,W,L,P:CLOSE:RETURN
280 REM  **  Zero values of T(x,y) **
290 FOR X1=1 TO 22:FOR Y1=1 TO 20:T(X1,Y1)=0:NEXT Y1:NEXT X1:RETURN
300 REM  ** Print to TOTALS/TXT  file **
310 OPEN"O",3,"TOTALS/TXT"
320 FOR X=1 TO P:GOSUB 340  :NEXT X:FOR X=21 TO 22:GOSUB 340  :NEXT X
330 CLOSE:RETURN
340 FOR Y=1 TO 20:PRINT#3,T(X,Y):NEXT Y:RETURN
350 REM  ** Input from TOTALS/TXT  file  **
360 OPEN"I",3,"TOTALS/TXT"
370 FOR X=1 TO P:GOSUB 390  :NEXT X:FOR X=21 TO 22:GOSUB 390  :NEXT X
380 CLOSE:RETURN
390 FOR Y=1 TO 20:INPUT#3,T(X,Y):NEXT Y:RETURN
400 REM  ** Print to PLAYERS/TXT  file **
410 OPEN"O",3,"PLAYERS/TXT"
420 FOR X=1 TO P:PRINT#3, P$(X);",";N$(X):NEXT X:CLOSE:RETURN
430 REM  ** Input from PLAYERS/TXT  file  **
440 OPEN"I",3,"PLAYERS/TXT"
450 FOR X=1 TO P:INPUT#3,P$(X),N$(X):NEXT X:CLOSE
460 P$(21)="Team Totals": N$(21)="-":P$(22)="Opponents": N$(22)="-"
470 RETURN
480 REM  ** Zero values of S(x)  **
490 FOR X1=1 TO 20:S(X1)=0:NEXT X1:RETURN
500 REM  ** Zero values of A(x)  **
510 FOR X1=1 TO 20:A(X1)=0:NEXT X1:RETURN
520 REM  ** Update 12 Statistics for players and opponents **
530 PRINT@449,"Field Goals Made"; : INPUT A(3)
540 PRINT@481,"Field Goals Attempted"; : INPUT A(4)
550 PRINT@513,"Free Throws Made"; : INPUT A(6)
560 PRINT@545,"Free Throws Attempted"; : INPUT A(7)
570 PRINT@577,"Offensive Rebounds"; : INPUT A(11)
580 PRINT@609,"Defensive Rebounds"; : INPUT A(12)
590 PRINT@641,"Turnovers"; : INPUT A(15)
600 PRINT@673,"Assists"; : INPUT A(16)
610 PRINT@705,"Draw Offensive Fouls"; : INPUT A(17)
620 PRINT@737,"Steals"; : INPUT A(18)
```

Listing continued

Listing continued

```
630 PRINT@759,"Blocked Shots"; : INPUT A(19)
640 PRINT@801,"Fouls"; : INPUT A(20): RETURN
650 REM  ** Sum statistics for each individual player **
660 FOR Y=2 TO 20:T(X,Y)=T(X,Y)+A(Y):S(Y)=S(Y)+A(Y):NEXT Y
670 IF T(X,4)=0 THEN T(X,5)=0 ELSE T(X,5)=T(X,3)/T(X,4)*100
680 IF T(X,7)=0 THEN T(X,8)=0 ELSE T(X,8)=T(X,6)/T(X,7)*100
690 T(X,9)=T(X,3)*2+T(X,6):T(X,13)=T(X,11)+T(X,12)
700 IF T(X,1)>0 THEN T(X,10)=T(X,9)/T(X,1):T(X,14)=T(X,13)/T(X,1)
710 IF A(4)>0 THEN A(5)=A(3)/A(4)*100
720 IF A(7)>0 THEN A(8)=A(6)/A(7)*100
730 A(9)=2*A(3)+A(6):A(10)=A(9):A(13)=A(11)+A(12):A(14)=A(13):RETURN
740 REM  ** Forms player file name from player number **
750 T$=STR$(X):T$=RIGHT$(T$,LEN(T$)-1):F$="PLAYER"+T$+"/TXT":RETURN
760 REM  ** Open Random - Access file  **
770 OPEN"R",1,F$,42
780 FIELD 1,2 AS QB$,2 AS FM$,2 AS FA$,4 AS FP$,2 AS FT$,2 AS F2$,4 AS F3$,2 AS F$
,2 AS SA$,2 AS OO$,2 AS DR$,2 AS TR$,2 AS RA$,2 AS TN$,2 AS AA$,2 AS DF$,2 AS ST$,
2 AS BS$,2 AS PO$:RETURN
790 REM  ** rset statistics in the buffer **
800 RSET QB$=MKI$(A(2)):RSET FM$=MKI$(A(3)):RSET FA$=MKI$(A(4))
810 RSET FP$=MKS$(A(5)):RSET FT$=MKI$(A(6)):RSET F2$=MKI$(A(7))
820 RSET F3$=MKS$(A(8)):RSET P$=MKI$(A(9)):RSET SA$=MKI$(A(10))
830 RSET OO$=MKI$(A(11)):RSET DR$=MKI$(A(12)):RSET TR$=MKI$(A(13))
840 RSET RA$=MKI$(A(14)):RSET TN$=MKI$(A(15)):RSET AA$=MKI$(A(16))
850 RSET DF$=MKI$(A(17)):RSET ST$=MKI$(A(18)):RSET BS$=MKI$(A(19))
860 RSET PO$=MKI$(A(20)):RETURN
870 REM  ** PUT INFO INTO R-A FILE **
880 PUT 1,G:CLOSE:RETURN
890 REM  ** Sum team totals **
900 S(1)=1:S(2)=4
910 IP S(4)>0 THEN S(5)=S(3)/S(4)*100
920 IF S(7)>0 THEN S(8)=S(6)/S(7)*100
930 S(9)=S(3)*2+S(6):S(10)=S(9):S(13)=S(11)+S(12):S(14)=S(13)
940 FOR Y=1 TO 20:T(21,Y)=T(21,Y)+S(Y):NEXT Y
950 IF T(21,4)>0 THEN T(21,5)=T(21,3)/T(21,4)*100
960 IF T(21,7)>0 THEN T(21,8)=T(21,6)/T(21,7)*100
970 IF G>0 THEN T(21,10)=T(21,9)/T(21,1):T(21,14)=T(21,13)/T(21,1)
980 SO=S(9):RETURN
990 REM  ** Change S( ) to A( )  **
1000 FOR Y=1 TO 20:A(Y)=S(Y):NEXT Y:RETURN
1010 REM  ** PRINT TO GAMES/TXT  FILE **
1020 IF G=1 THEN OPEN"O",3,"GAMES/TXT" ELSE OPEN"E",3,"GAMES/TXT"
1030 PRINT#3, O$;",";D$;",";H$;",";SU;ST:CLOSE:RETURN
1040 REM  ** Lprint Heading for printouts - first 4 lines **
1050 LPRINT" ";T$=S$+" Basketball Statistics":GOSUB 1110
1060 LPRINT TAB(T) T$:LPRINT TAB(50) Y$:LPRINT " ":RETURN
1070 REM  ** Lprint lines 5 and 6 of most printouts **
1080 T$="Coach: "+C$+"     Won:"+STR$(W)+"    Lost:"+STR$(L):GOSUB 1110
1090 LPRINT TAB(T) T$:LPRINT " ":RETURN
1100 REM  ** Center Titles **
1110 T=LEN(T$):T=INT((109-T)/2):RETURN
1120 REM  ** Assign temporary variables to be printed **
1130 T$="Sco. Off Def Tot Reb. Trn     Drw     Blk"
1140 LPRINT TAB(61) "     "; T$:RETURN
1150 T$=" # Player":RETURN
1160 T$="G Qtr FGM  FGA  PG % FTM FTA  FT % Pts. Avg. Reb Reb Reb Avg. Ovr  A  Fou
 Stl Sht  P":RETURN
1170 T$="-- "+STRING$(21,"-"):RETURN
1180 T$=" -- --- --- ---- ----- --- --- ----- ---- ---- --- --- --- ---- --- --- -
-- --- --- ---":RETURN
1190 T$="##":RETURN
1200 T$=" #                    #":RETURN
1210 T$="##":RETURN
1220 T$=" ###":RETURN
1230 T$=" ###.#":RETURN
1240 T$=" ##.#":RETURN
1250 T$=" #       #":RETURN
1260 T$=" #             #":RETURN
1270 T$="G#   Date    Opponent       H/A  ":RETURN
1280 GOSUB 1180 :T$="-- -------- "+STRING$(14,"-")+" ----"+RIGHT$(T$,80):RETURN
1290 T$=" ####":RETURN
1300 T$="##    Opponent: #            #  #":RETURN
1310 REM  ***** Insert Paper  message  *****
1320 CLS:PRINT"Do you have an Epson Printer? ":GOSUB 210  :IF T$="N" THEN 1350
1330 PRINT"Do you want compressed printing? ":GOSUB 210
1340 IP T$="Y" THEN LPRINT CHR$(15):P2=1
1350 PRINT:PRINT"Please insert paper in your printer.":PRINT
1360 PRINT"Now Press  <ENTER> when you are ready to print.";:GOSUB 170
1370 PRINT:PRINT"Statistics are now being printed."
1380 PRINT:PRINT"Please do NOT remove paper until the menu reappears on the" :
     PRINT"screen.":RETURN
1390 REM  *****  Change it?  *****
1400 PRINT:PRINT"Do you wish to change it (Y/N)?";:GOSUB 210  :RETURN
1410 REM ***** Print Individual's statistics on Screen *****
1420 P@INT@64,"Player:  ";P$(PN);
1430 PRINT@193,"Quarters: ";A(2);
1440 PRINT@257,CHR$(31);"Field Goals Made:";A(3);
1450 PRINT@289,"Field Goals Attempted:";A(4)
1460 PRINT@321,"Free Throws Made:";A(6);
1470 PRINT@353,"Free Throws Attempted:";A(7);
1480 PRINT@385,"Offensive Rebounds:";A(11);
1490 PRINT@417,"Defensive Rebounds:";A(12);
1500 PRINT@449,"Turnovers:";A(15);
1510 PRINT@481,"Assists:";A(16);
1520 PRINT@513,"Draw Offensive Fouls:";A(17);
1530 PRINT@545,"Steals:";A(18);
1540 PRINT@577,"Blocked Shots:";A(19);
1550 PRINT@609,"Fouls:";A(20);:RETURN
1560 CLS:OPEN"I",3,"GAMES/TXT"
1570 FOR Y1=1 TO G:INPUT#3,O$,D$,H$,SU,ST:PRINT Y1;O$;" - ";H$,:NEXT Y1
1580 CLOSE:RETURN
1590 IF P2=1 THEN LPRINT CHR$(18):RETURN
1600 RETURN
```

Listing continued on p. 134

WINDOW SCREENS

by Glen E. Sparks

Simultaneously display multiple windows of text or graphics with BasicG and a high-resolution board.

Everywhere you look these days, you see programs that use windows and pull-down menus. Everywhere, that is, except in Model III/4 Basic. But you Basic programmers needn't feel left out—with a high-resolution graphics board and BasicG, you can simulate a windowing environment in your own programs. You get the ability to display several windows of data or graphics simultaneously, manipulate pull-down menus, and create some dramatic animated graphics.

I'll explain how the windowing technique works, and how to use the commands available to you. For illustration, I'll also present a complete application program, a pie chart generator, that uses four windows and a pull-down menu.

RAM with a View

The key to creating windows on the Models III and 4 is BasicG's View command. (BasicG is the graphics Basic that comes with Radio Shack's high-resolution board.) Essentially, this command makes your computer act as though a portion of the screen, called a viewport or window, is in fact the entire screen. Therefore, you can erase or change a window without affecting the rest of the screen. You can write to, draw on, or clear only the last viewport you defined.

When you clear a viewport, it erases everything underneath. You can define and clear viewports all day long if you want, stacking each new viewport on top of the last. Program Listing 1, Sinewave, and Program Listing 2, Prism Ring, create three-dimensional graphics effects using this technique (see Photos 1 and 2). You can also divide the screen and display windows next to each other.

Program Listing 3, Viewport, illustrates using windows to display data. It paints the entire screen with a pattern, defines the center of the screen as a viewport, clears the viewport, and displays a message there. Then it repeats the process for a second viewport below and to the right of the first (see Photo 3).

Notice that when text reaches the border of a viewport, it wraps around, just as it normally does at the edge of the full-width screen. Also notice that the two viewports aren't the same size. This means the text wraps around sooner on one than on the other. In your own programs, you'd have to include a subroutine to check the size of the viewport and split words logically where needed.

In BasicG, you use the GLOCATE(X,Y),0 statement instead of PRINT@ to display text at a specific place on the screen. GLOCATE defines the coordinates, and the command PRINT#-3 does the printing. The syntax is the same whether you're printing over the entire screen or in viewports. However, once you've defined a viewport, the coordinate system becomes relative to that viewport. In Listing 3, even though the windows aren't in the upper left-hand corner of the screen, the windows' upper left coordinates are 1,1 (line 210). You can therefore use the same subroutine to put data in any window by addressing the same X,Y coordinates.

There's one hitch to all this: Because the computer treats a viewport as an entire screen, you get an error message if you try to write past the parameters of the last viewport you defined. To avoid this problem, I suggest you redefine the entire screen as a viewport when you exit a program that uses viewports.

Overwriting Concerns

As I mentioned above, defining a new window destroys anything under it on the screen. However, if you've seen commercial programs that use windows or pull-down menus (menus that slide down, covering a portion of the screen without destroying it), you probably noticed that the areas underneath appear to be intact. You can do the same trick with BasicG's Get and Put commands. (In this case, Get and Put don't work as they do with random-access files; they're special BasicG graphics commands.)

The statement GET(X1,X1)-(X2,Y2), VAR% stores a section of the screen in an array (VAR%) that you dimension earlier. Unlike the View command, this doesn't affect the screen. Also unlike View, this is a memory-hungry command. You might need a dimensioned array of 2.5K to store a quarter of the screen.

Once you store a section of screen, however, you can use the Put command to display it over and over again with little memory penalty. This is useful for pull-down menus or for storing an area a viewport overwrites. Program Listing 4, Circle, shows how this works; it draws a circle inside a box in the upper right corner of the screen, paints over it, and then restores it. Substitute PRESET for PSET in line 80 to restore the image in inverse video.

You can use this technique to restore a portion of the screen you're going to overwrite with a viewport. Figure out how much area you'll overwrite and dimension an array large enough to store it. The appropriate formula appears in your graphics Basic manual. In a Get statement, save a section of screen comparable to the area you want covered, then use a Put statement to restore it after you use the window.

If you're working with viewports, you'll have to redefine the entire screen as a viewport or redefine the section where you're restoring your data. Otherwise, you'll get an error message if you try to write past the confines of the last viewport.

This might sound complicated, but it's easier done than said. Likewise pull-down menus. You simply design a menu and GLOCATE it to the screen much as you would a block of text on the normal screen. Save the menu with the Get command in an array large enough to hold it. Erase the graphics screen and proceed with your program.

When you need a pull-down menu, save the area that the menu will cover in another array with another Get statement. The two arrays are the same size. Use the Put command with the menu array to display the menu on the screen. After the menu's INKEY$ routine, replace the original section of the screen and erase the menu at the same time by putting the second array back where you put the menu. The menu shrinks away as if it had never been there.

Using the methods I've described, you can write your own window programs with pull-down menus. Obviously, those monster multiwindow programs for MS-DOS machines aren't written in Basic, but the logic is the same.

LOAD 80

System Requirements

Models III and 4
48K RAM
BasicG
High-resolution board
Printer optional

Pie Are Not Square

Windows is a pie chart program that puts the principles described above to work (see Program Listing 5). You can display up to four pie charts at a time on its four independent windows. A pull-down menu lets you manipulate the display.

When you run Windows, you'll see the pull-down menu form on the screen and quickly disappear. The program saves it in a Get array for later use. The input routine now prompts you for the title of a pie chart; the prompt appears on the non-graphics screen (in all, you have five screens—the normal screen plus four graphics windows). Your title can be any combination of numbers and letters up to 15 characters long.

Next, Windows prompts you for the period of time the pie chart covers; the same input restrictions apply here. Then you specify how many entries, or accounts, you want to chart. The limit, nine entries, is governed by the windows' size.

Now you choose the window where you want to display your data, that is, the chart's title and raw figures (see Photo 4). Type in a number from 1 to 4. Window 1 is the screen's upper left corner, 2 is lower left, 3 is upper right, and 4 is lower right.

Next, you're prompted for the number of a window for the pie chart itself; again, type in a number from 1–4. The program does no error-checking here to see if you type in the same window number for both your data and the chart. If you indicate the same number, Windows will display the data, then immediately erase it and display the corresponding pie chart. I set up the program this way so you can display four pie charts at once, one in each window.

The next block of prompts repeats for each account you indicated. For each account, you type in an account name and amount. The name can be up to nine characters long. The amount's upper limit is 99,999.99. When you type in numeric data, don't use commas, since Windows reads them as delimiters.

If you need larger amounts, you can reformat the program's Print Using statements. However, you'll have less room for the account name if you do so.

The pie chart algorithm has a mechanism that excludes low amounts if the difference between amounts is great. This is necessary for clarity's sake—some slices would be comparatively too small to chart. All amounts you input are included in the total column, however. Since pie slices are numbered, you'll be able to tell which amounts didn't chart.

After you type in the last amount, Windows automatically goes to the graphics screen, draws all four windows, and displays the data and chart in the windows you specified (see Photo 4).

At this point, you can call up the pull-down menu by tapping the spacebar. It

Delete line 160

Change line 580 to:
580 LPRINT CHR$(27):CHR$(20):
SYSTEM "GPRT2":RETURN

Insert line 615:
615 SCREEN 1

Insert line 905:
905 SCREEN 0

Figure. Modifications to Program Listing 5 for the Model 4.

appears in the middle of the screen as if it were on a shade that someone was pulling down (see Photo 5). Tap the spacebar again and the menu disappears.

To invoke a menu option, press the number key corresponding to the menu number on the screen. You can erase a window, dump the screen to the printer, or go back to the input subroutine. You don't have to erase a window that you plan to overwrite with a new pie chart or data; Windows does it automatically when you choose that window number during data entry.

When you send your report to the printer, you exit to BasicG in Model 4 mode or exit to TRSDOS in Model III mode.

To run Windows on a Model 4, you must modify Listing 5 as shown in the Figure.

Charting Your Own Course

The power of a window environment lies in its ability to display different data or types of data at the same time. I set up the windows in this program for visual effect and to show that a window's placement and size is arbitrary and not restricted to any one layout. Your requirements might suggest only two windows or more than four.

I left the input section relatively simple. You have more than enough memory left over to add disk I/O routines for VisiCalc DIF files or data base management interfaces. You should have no trouble finding ways to tailor Windows to your own specifications. ■

Glen E. Sparks is a programmer and a member of the Dearborn, MI, user's group. You can write to him at 6186 Custer, S. Rockwood, MI 48179.

Related Articles

Rowell, Dave, "Sifting Through GW-Basic," August, 1985, p. 46. A GW-Basic tutorial that covers the View command.

Also, see this month's installment of Dave's MS-DOS column, p. 92, for a Model 1000 conversion of the Sinewave program.

Photo 1. Sinewave's display.

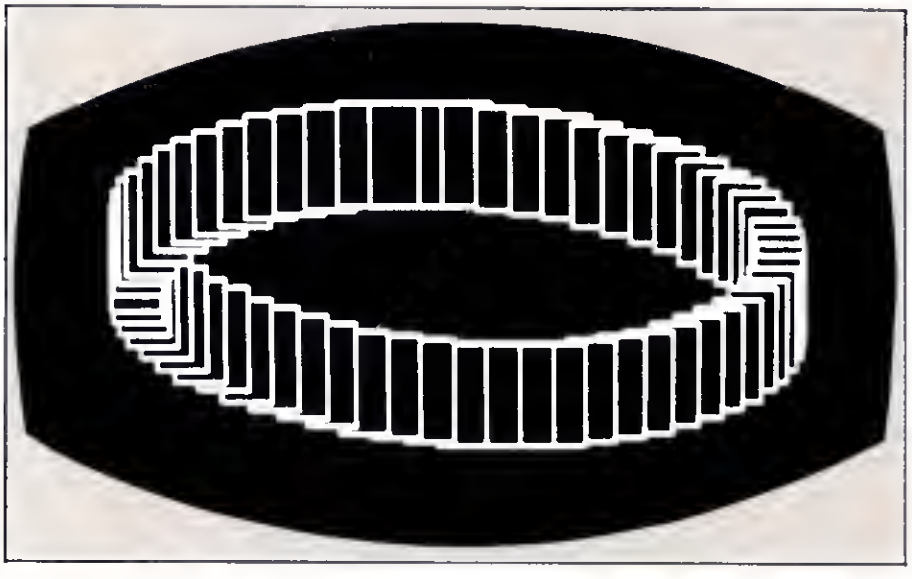


Photo 2. Prism Ring's display.

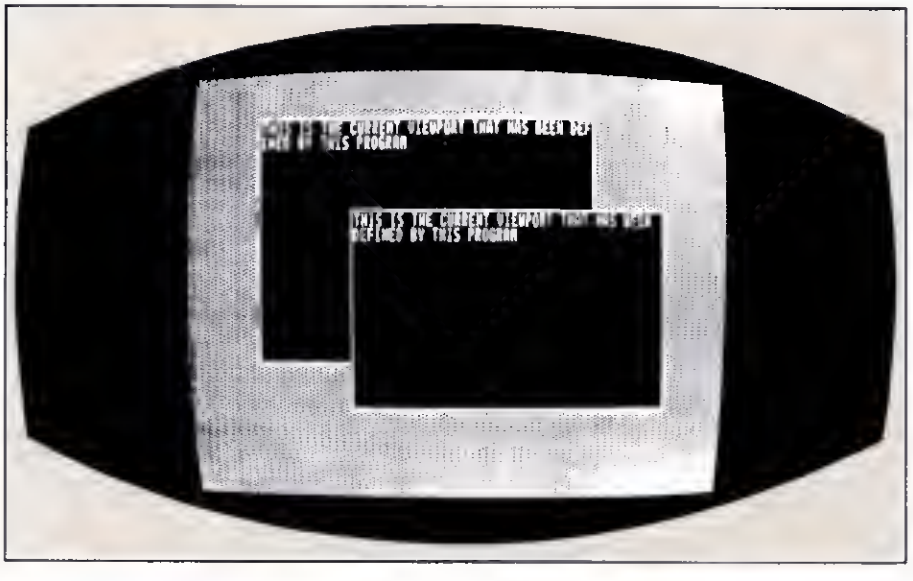


Photo 3. Viewport's display.

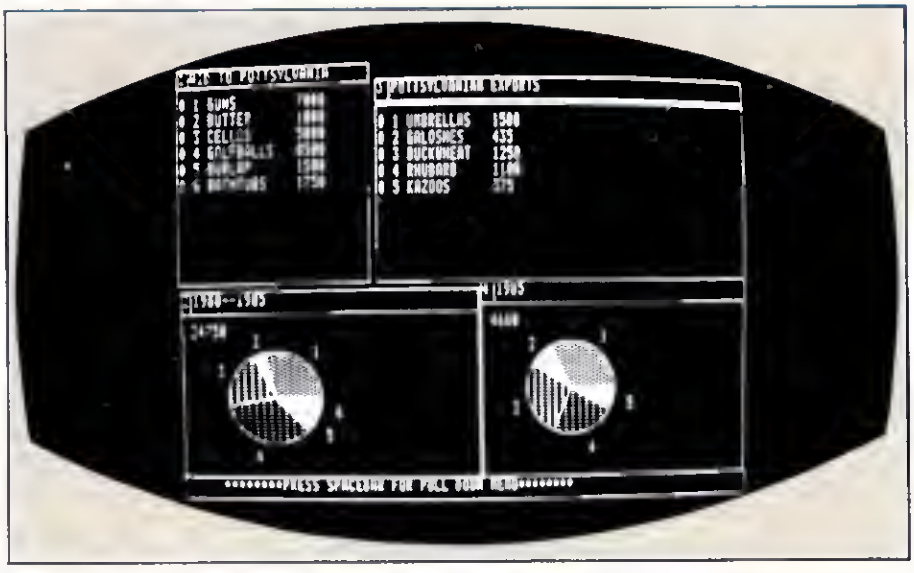


Photo 4. Windows' data displays and pie charts.

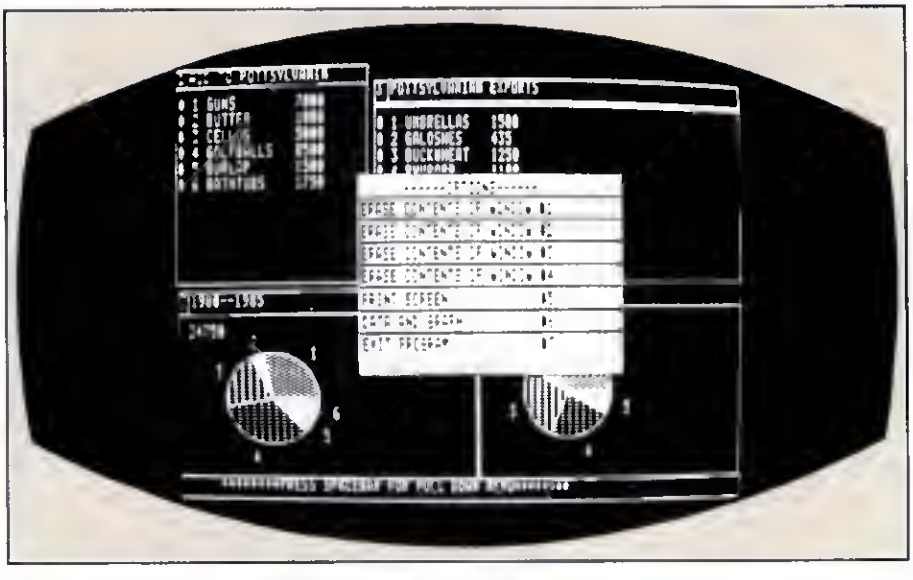


Photo 5. Windows' pull-down menu.

Program Listing 1. Sinewave.

```
10 ' SINE WAVE DESCENDS DOWN SCREEN
20 VIEW(0,0)-(639,239):CLR:SCREEN 0 'set entire graphics screen to
 viewport and clear it
30 C=55  'lower the number, the flatter the wave
40 J=0:Z1=5:Z2=.9:A=0:B=12
50 FOR X=A TO B STEP .15
60 X1=20*X:Y=SIN(X):Y1=139-C*(Y+1) 'sin wave algorithm--plot where
boxes are to be on screen
70 IF C<0 THEN    'if C<0 then error-send to endless loop or begin
   prog over
00 IF J>0 THEN C=C-.04 'increasing minus off C increases spaces
90 IF J>0 THEN X1=X1+2:Y1=Y1-.01
100 IF X1<0 THEN X1=X1*-.1
110 VIEW(X1,Y1)-(X1+Z1,Y1+Z1),,1:CLR 'draw actual viewport (box)
    and clear its contents thus removing hidden lines
120 Z1=Z1+Z2:NEXT 'make boxes in wave larger to midpoint of wave
130 Z2=-Z2:J=J+1 'if midpoint reached then make boxes smaller-J is
    counting variable for loop
140 IF J=2 THEN 160 'if second half wave made-go to acreen holding
    loop
150 A=12:B=23:GOTO 50  'midpoint starting variables -execute first
    part of program with new values to make second half of wave
160 IF INKEY$="" THEN 160
```

Continued on p. 138 *End*

And now, a couple of words about high-quality TRS-80 software at a very low price:

FREE Bonus Program this issue!

LOAD 80

Utilities, tutorials, home and hobby applications from 80 Micro.

If you've been shopping for software lately, you've discovered that new car buyers aren't the only ones who experience "sticker shock".

For the price of one commercial program, you can fill your gas tank at least three times. Or treat a friend to dinner. Or buy a year's worth of Sunday papers.

But with *Load 80* software, you can spend a lot less and *still* wind up with hundreds of dollars worth of outstanding TRS-80 programs every month.

On every *Load 80* cassette or disk (your choice!), you'll get more than a dozen "ready to run" programs listed in *80 Micro*... tutorials, utilities, games, word-processing, and much more.

Build a versatile software library, quickly and economically. Past issues have included programs such as:

NovaCalc

...a full-featured Model I/III spreadsheet with all the capabilities offered on more expensive commercial products.

Easydata

...a 200-record data base manager for fast information from your Model I/III/4.

Grade-A Graphics

...a deluxe Model III graphics editor that's loaded with options!

And to enjoy your favorite program, all you have to do is "load 80" into your computer. It's that simple. No keyboarding, no debugging. You get complete loading instructions, but should you need assistance, the *Load 80* and *80 Micro* technical editors will be glad to answer your questions.

Don't let software sticker prices stop you from building a top-notch library. Get a variety of winning programs, for a fraction of the cost, with *Load 80* cassettes or disks.

To order by the month or by the year, simply complete the coupon and drop it in the mail with payment.

For Faster Service, call

☎ 1-800-258-5473

(In NH, please dial 1-924-9471)

Yes. ***I want to build an exciting and economical software library with Load 80.*** Please send me:

☐ 1 year of Load 80 on disk for $199.97
☐ 1 year of Load 80 on cassette for $99.97
☐ This month's Load 80 disk for $21.47
☐ This month's Load 80 cassette for $11.47

☐ Check/MO ☐ MC ☐ VISA ☐ AE

Card # _____ Exp. Date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

Prices include postage and handling. Foreign airmail, please add $1.90 per item or $25.00 per subscription. US funds drawn on US banks only.

Load 80 • 80 Pine Street • Peterborough, NH 03458 12-85

Interrupt Anytime

Twelve programmable interrupts—at two speeds—for Model III TRSDOS programmers.

An interrupt is a hot line to your computer's mission control. It breaks into an executing program so the processor can accomplish some other task, and it works so fast that the computer doesn't miss a beat. Interrupts are especially valuable because they're invisible to the user.

Most Model III programmers can't take advantage of interrupts because TRSDOS 1.3 doesn't use them, except to update the internal clock. Program Listing 1, Break In, gives TRSDOS 1.3 complete interrupt-handling capability. Once you install the program, you can run up to 12 interrupt-driven tasks simultaneously.

Interrupts lend themselves to countless applications: type-ahead routines, printer spoolers, INKEY routines, and so on. I'll tell you how to write your own interrupt routines, and I'll provide you with a sample routine that adds a Scroll command to Basic.

Clock Work

The Model III's clock hardware sends a special signal that interrupts the computer's central processing unit (CPU) so software in ROM can update the clock. These interruptions occur extremely fast—about every 33.333 milliseconds (ms.).

When a clock interrupt occurs, control jumps to location 4012 hexadecimal (hex). Under TRSDOS 1.3, location 4012 hex simply redirects the computer to 3018 hex, which updates the clock's time and takes care of other necessary chores. By changing the instruction at 4012, you can direct the computer to one of your own routines. For example, you could set up a program to trace what location in memory the CPU executes, or you could write a program that sounds an alarm at a certain time.

LOAD 80

System Requirements
Model III
48K RAM
TRSDOS 1.3
Assembly language
Editor/assembler

Program Listing 1. Break In.

```
00150 ;
00160          ORG     0FE00H
00170 START    CALL    457             ;CLEAR SCREEN
00180          LD      HL,MSG1         ;GET OPTION MESSAGE
00190          CALL    539             ;DISPLAY A LINE
00200 WAIT     CALL    73              ;WAIT FOR A KEYPRESS
00210          CP      'E'             ;WAS ENABLE OPTION CHOSEN
00220          JR      Z,ENABLE        ;IF SO, ENABLE IT
00230          CP      'D'             ;DISABLE OPTION CHOSE?
00240          JR      NZ,WAIT         ;IF NOT, LOOP AGAIN
00250          LD      A,15            ;TURN CURSOR OFF CHAR.
00260          CALL    51              ;TURN CURSOR OFF
00270          LD      A,13            ;DISPLAY A CARRIAGE RETRN
00280          CALL    51
00290          LD      HL,MSG3         ;GET "DISABLED" MESSAGE
00300          CALL    539             ;PRINT IT
00310          LD      HL,3018H        ;ORIGINAL ROUTINE ADDRESS
00320          DI                      ;DISABLE INTERRUPTS
00330          LD      (4013H),HL      ;RESTORE ORIGINAL ROUTINE
00340          EI                      ;ENABLE INTERRUPTS
00350          LD      HL,0FFFFH       ;RESET HIGH RAM MEMORY
00360          LD      (4411H),HL      ;LOCATION.
00370          JP      402DH           ;EXIT TO TRSDOS READY
00380 ENABLE   LD      A,15            ;CURSOR OFF CHARACTER
00390          CALL    51              ;TURN CURSOR OFF
00400          LD      A,13            ;DISPLAY A CARRIAGE RETRN
00410          CALL    51
00420          LD      HL,MSG2         ;GET "ENABLED" MESSAGE
00430          CALL    539             ;DISPLAY IT
00440          DI                      ;DISABLE INTERRUPTS
00450          LD      HL,START2       ;INTERRUPT HANDLER
00460          LD      (4013H),HL      ;INSTALL INTERRUPT HANDLER
00470          LD      HL,START-1      ;VALUE TO PROTECT PROGRAM
00480          LD      (4411H),HL      ;PROTECT THE PROGRAM
00490          EI                      ;ENABLE INTERRUPTS
00500          JP      402DH           ;EXIT - EVERYTHING WORKS
00510 MSG1     DEFM    '<E>nable or <D>isable the clock interrupt routine? '
00520          DEFB    14
00530          DEFB    3
00540 MSG2     DEFM    'The clock interrupt routine has been ENABLED!'
00550          DEFB    0DH
00560 MSG3     DEFM    'The clock interrupt routine has been DISABLED!'
00570          DEFB    0DH
00580          ORG     0FF00H          ;PROGRAM RESIDES IN HIMEM
00590 SLOW1    DEFW    DEFALT          ;VECTORS FOR SLOW INT'S.
00600 SLOW2    DEFW    DEFALT
00610 SLOW3    DEFW    DEFALT
00620 SLOW4    DEFW    DEFALT
00630 SLOW5    DEFW    DEFALT
00640 SLOW6    DEFW    DEFALT
00650 SLOW7    DEFW    DEFALT
00660 SLOW8    DEFW    DEFALT
00670 FAST1    DEFW    DEFALT          ;VECTORS FOR FAST INT'S.
00680 FAST2    DEFW    DEFALT
00690 FAST3    DEFW    DEFALT
00700 FAST4    DEFW    DEFALT
00710 ;
00720 ; Start of interrupt handling routine
00730 ;
00740 START2   PUSH    DE              ;SAVE REGISTERS
00750          PUSH    AF
00760          PUSH    HL
00770          PUSH    BC
00780          PUSH    IX
00790          PUSH    IY
00800          LD      DE,START3       ;RETURN LOCATION
00810          PUSH    DE              ;SAVE RETURN LOCATION
00820          LD      A,8             ;THE FOLLOWING EXECUTES
00830          CALL    FAST            ;A FAST 33.33 MILLISECOND
00840          LD      A,9             ;USER-DEFINED INTERRUPT.
00850          CALL    FAST
00860          LD      A,10
00870          CALL    FAST
00880          LD      A,11
00890          CALL    FAST
00900          LD      HL,TIMER        ;INCREMENT THE 33.333
00910          INC     (HL)            ;MILLISECOND COUNTER
```

Listing 1 continued

ANYONE CAN USE OUR SOFTWARE!

MAIL PAC II and CHECKING PLUS represent a new generation of computer software. Our software doesn't even need a manual! Anyone can quickly learn to use either package by following simple, explicit on-screen instructions. But, for those of you who insist -- we've included a complete user's manual as well.

MAIL PAC II FEATURES:

High Capacity --
Stores from one to one million names in Zip Code, Numerical or Alphabetical order. The only limitation is your disk storage space.

Flexibility --
Prints your mailing list on 1, 2, 3, or 4-across labels (with up to 5 user-defined lines on each label) or as a compact, user-designed directory. The record length is completely user-defined, and each field within each of your records is completely variable (allowing storage of any number of characters for any particular address entry). Devote fields to telephone numbers, codes, or even special messages related to each particular name on file.

Powerful --
A built-in word processor allows you to create personalized form letters for each address on your list (or just a particular group of addresses).

Ease of Use --
Create new mailing lists, review existing lists, handle changes of address, delete cancelled names, sort lists, and purge duplicate names from your files. Complete on-screen instructions tell you in plain english exactly how to accomplish all of these tasks.

MAIL PAC II $99.95

CHECKING PLUS FEATURES:

Complete Check Register --
Checking Plus stores your entire check register in a disk file, and then uses the data to balance your account, track your expansas, and help you make budget projections. Review the entire checkbook, enter checks, deposits, fees and adjustments, mark outstanding checks when paid, and get an instant cash balance at any time. All data can be viewed onscreen or printed out in report form.

Tax Preparation --
Storage of monthly and yearly totals and other important information aids in income tax preparation, for your personal use or for your accountant.

Handles the Details --
Store names and addresses for frequently written checks, then print checks to fit standard window envelopes, eliminating the need for extra typing. You can even store any comment, explanation or other message (up to 255 characters) related to a particular check.

Automatic Monthly Bill Payments --
Enter amounts and names of payees for all of your fixed-cost monthly payments, and then sit back while the system automatically prints checks.

CHECKING PLUS $99.95

Both Programs Require 48K And At Least One Disk Drive (2 Drives For Higher Storage Capacities). And All Hard Disk Systems Are Supported.

FREE--TRS-80 Mod I,III & 4 programs supplied on DosPlus (minimum system). Complete DosPlus also available.

FOR YOUR TRS-80™ • APPLE™ • IBM PC™ • NEC™ • OSBORNE™ • XEROX™ • KAYPRO™ • TELEVIDEO™ • ZENITH™ • SANYO™
DEC™ • TI PROFESSIONAL COMPUTER™ • SUPERBRAIN JR.™ • EPSON™ • Any CP/M™ Computer

CP/M-based Computers must be equipped with Microsoft BASIC (MBASIC or BASIC-80)

OUR SOFTWARE CATALOG

H & E Computronics, Inc., has mailed more than 1 million software catalogs since 1978. Send $2 for our new 64-page catalog today! (We also send you our catalog FREE with every order) DEALER INQUIRIES WELCOME

30-DAY MONEY BACK GUARANTEE

*** ALL PRICES & SPECIFICATIONS SUBJECT TO CHANGE ***
DELIVERY SUBJECT TO AVAILABILITY

COMPUTRONICS

50 N. PASCACK RD., SPRING VALLEY, N.Y. 10977

24 HOUR ORDER LINE
(914) 425-1535

ADD $3.00 FOR SHIPPING IN UPS AREAS
ADD $4.00 FOR C.O.D. OR NON-UPS AREAS
ADD $5.00 TO CANADA & MEXICO
ADD PROPER POSTAGE OUTSIDE OF U.S. CANADA & MEXICO

NEW TOLL-FREE ORDER LINE
(OUTSIDE OF N.Y. STATE)
(800) 431-2818

VISA MasterCard

Circle 9 on Reader Service card.

Listing 1 continued

```
00920           LD      A,(HL)          ;GET COUNTER VALUE
00930           AND     7               ;# OF SLOW INT. TO RUN
00940 FAST      RLCA                    ;DETERMINE INTPT LOCATION
00950           LD      L,A             ;HL = ADDRESS OF NEXT
00960           LD      H,0FFH          ;INTERRUPT TO EXECUTE
00970           LD      E,(HL)          ;DE = LOCATION OF THE
00980           INC     L               ;STARTING 2 BYTES
00990           LD      D,(HL)          ;POINTING TO YOUR INTRUPT
01000           PUSH    DE              ;COPY DE INTO IX
01010           POP     IX              ;IX CONTAINS A COPY OF DE
01020           EX      DE,HL           ;EXCHANGE DE AND HL
01030           LD      E,(HL)          ;DE = ACTUAL LOC. OF YOUR
01040           INC     HL              ;INTERRUPT ROUTINE
01050           LD      D,(HL)
01060           EX      DE,HL           ;EXCHANGE DE AND HL AGAIN
01070           JP      (HL)            ;EXECUTE YOUR ROUTINE
01080 REMOVE    LD      DE,DEFALT       ;INT. VECTOR RESET VALUE
01090 ADD       CP      12              ;IS TASK # > 12?
01100           RET     NC              ;RETURN IF MORE THAN 12
01110           RLCA                    ;GET OFFSET VALUE
01120           LD      L,A             ;HL = TASK LOCATION IN
01130           LD      H,0FFH          ;THE INTERRUPT TASK TABLE
01140           DI                      ;DISABLE INTERRUPTS
01150           LD      (HL),E          ;ADD THE TASK TO TABLE
01160           INC     HL              ;BUMP THE POINTER
01170           LD      (HL),D          ;INSTALLATION COMPLETE
01180           EI                      ;ENABLE INTERRUPTS AGAIN
01190 CONST     RET                     ;RETURN TO THE CALLER
01200 TIMER     DEFS    6               ;33.333 MS COUNTER
01210 DEFALT    DEFW    CONST           ;DEFAULT INT. LOCATION
01220 START3    POP     IX              ;RESTORE REGISTERS
01230           POP     BC
01240           POP     HL
01250           POP     AF
01260           POP     DE
01270           JP      38108           ;CONTINUE CLOCK HANDLER
01280           END     START
```

End

Program Listing 2. Demo.

```
00150 ;
00170           ORG     0FD90H          ;SI-RAM INTERRUPT
00180 ADDTSK    EQU     0FF52H          ;ADD A TASK SUBROUTINE
00190 REMTSK    EQU     0FF4FH          ;REMOVE A TASK SUBROUTINE
00200 TIMER     EQU     0FF5FH          ;LOCATION OF 33.3MS TIMER
00210 ;
00220 ; Interrupt installation procedure follows below
00230 ;
00240 START     CALL    457             ;CLEAR SCREEN
00250           LD      HL,MSG1         ;GET MESSAGE
00260           CALL    539             ;PRINT THE MESSAGE
00270           LD      DE,MAIN         ;POINTER OF INT. ROUTINE
00280           LD      A,0             ;SLOT ASSIGNMENT 0.
00290           CALL    ADDTSK          ;ADD THE TASK TO TABLE
00300           LD      HL,0FD0FH       ;SET SI-REMORT BELOW'
00310           LD      (44118),HL      ;PROGRAM TO PROTECT IT
00320           JP      402DH           ;RETURN TO TRSDOS
00330 MSG1      DEFM    'Interrupt ON!'
00340           DEFB    0DH
00350 ;
00360 ; This is the actual interrupt handling routine. It is
00370 ; executed every 267.67 milliseconds, or so.
00380 ;
00390 MAIN      DEFM    MAIN2           ;POINT TO MAIN ROUTINE
00400 MAIN2     LD      A,(15360+63)    ;GET CHARACTER ON SCREEN
00410           CP      '='             ;IF CHAR. AN EQUAL SIGN?
00420           JR      Z,CHANGE        ;IF IT IS, CHANGE IT
00430           LD      A,'='           ;IF IT ISN'T, THEN
00440           LD      (15360+63),A    ;CHANGE IT TO ONE
00450           RET                     ;RETURN FROM INTERRUPT
00460 CHANGE    LD      A,'-'           ;CHANGE TO A DASH
00470           LD      (15360+63),A    ;CHANGE IT ON SCREEN
00480           RET                     ;RETURN FROM INTERRUPT
00490           END     START
```

End

Program Listing 3. Scroll.

```
00170 ;
00180 SCREEN    EQU     15360
00190 DSPLY     EQU     539
00200 ;
00210 ;Start of Initialization Procedure
00220 ;
00230           ORG     0F200H          ;INITIALIZATION AREA
00240 STARTR    EI                      ;ENABLE INTERRUPTS
00250           LD      A,(TIMER)       ;GET 33.33 MS TIMER COUNT
00260           CP      0               ;IS THE INTERRUPT WORKING?
00270           JR      NZ,STRT2        ;IF SO, FINISH INIT.
00280           LD      BC,25000        ;SET COUNTER TO 25000
00290 LPSTRT    DEC     BC              ;DECREMENT COUNTER
00300           LD      A,B             ;GET COUNT
00310           OR      C               ;IS THE TIMER EQUAL TO 0?
00320           CP      0               ;FINISHED THE LOOP?
00330           JR      NZ,LPSTRT       ;IF NOT, CONTINUE.
00340           LD      A,(TIMER)       ;GET TIMER COUNT
00350           CP      0               ;STILL A ZERO?
00360           JR      NZ,STRT2        ;IF NOT, THEN FINISH INIT
```

Listing 3 continued

Address (hex)	Description
4012	Hook vector for interrupts. ROM hooks there every 33.333 ms. or so.
4411	A 2-byte area storing the highest usable memory location.
3018	ROM's interrupt-handling routine.
FF52	ADDTSK subroutine. This adds a task to the interrupt task table. DE = TCB address, A = slot assignment.
FF4F	REMTSK subroutine. This removes a task from the interrupt task table. A = slot assignment to remove.
FF5F	Timer location. Timer is a 33.333 ms. counter incremented about every 33.333 ms.
FF00–FF17	Interrupt task table area. FF00–FF0F are low-speed interrupts; FF10–FF17 are high-speed ones.

Table 1. Break In's routines.

Break In gives you four interrupts at 33 ms. Because this might be too fast for some applications, its remaining eight interrupts occur at a low speed of about 267.67 ms. With Break In activated, you can run up to 12 routines at the same time without really affecting the clock's time.

When TRSDOS turns off the clock interrupt, as it does for disk accesses, routines using the clock won't execute until TRSDOS turns the clock back on. Therefore, TRSDOS's clock isn't always accurate.

Installment Plan

Break In controls your interrupt-driven routines, called tasks, via a task table containing the addresses of 12 interrupt slots. The program contains all the routines you need to add or remove a task from the task table. Table 1 shows these routines' addresses and requirements, along with some other locations worth noting. TRSDOS increments the value of a special 1-byte location, called the timer, by 1 every 33.333 ms. You could use it, for example, as a seed value for a random number generator, since it constantly changes.

Once you assemble Listing 1 to disk, Break In takes only a few seconds to install. When you run the program by typing in its /CMD file name, it asks whether you want to enable or disable the clock

interrupt routine. If you want to run interrupt-related programs, press the "E" key. If you've already enabled the routine and want to disable it, press the "D" key.

Enabling the routine activates 12 interrupts so they're ready to run your tasks. It also protects your program by setting the high-memory bytes at 4411 and 4412 hex to point to the first byte below the interrupt task table. This protects the table and the accompanying code, except in programs that erase all memory regardless of the high-memory setting.

Once you enable the interrupt-handling routine, don't use TRSDOS's Clear command, which clears all memory from 5600–FFFF hex. Before using Clear, run Listing 1 again and disable the routine.

Pressing the "D" key turns off the interrupt-handling routine and stops the 12 interrupts' operation. It resets the high-memory locations to point to the top of memory (FFFF hex).

After you run Break In, it returns you to TRSDOS. Now you can load your own task driver software. Program Listing 2, Demo, is a demonstration routine; it lets you see exactly when the interrupt executes. To run the routine, assemble Listing 2 to disk and, with Break In enabled, type in Listing 2's /CMD file name.

The message "Interrupt on!" appears on your screen. You should see a hyphen and an equals sign alternate in the screen's upper right-hand corner. The character changes each time the interrupt executes.

Demo gives you a low-speed interrupt, executing every 267.67 ms. or so. To see what a fast interrupt looks like, change line 280 of Listing 2 to LD A,8. This assigns the task to the first high-speed slot, so it operates every 33.333 ms. Now run Demo again and watch what happens. The hyphen and equals sign should alternate extremely fast.

Again, don't use the Clear command, unless you want your computer to crash. If the characters stop alternating, it means you're running a program that disables interrupts. Going back to TRSDOS Ready should enable them again.

Driver Education

Listing 2 illustrates how to write your own task driver. To add a task to Break In's task table, you must meet the following requirements. Register DE must point to a 2-byte address called the task control block (TCB), which contains the address of the driver's entry point. Register A must contain the interrupt's slot assignment: Slots zero to 7 represent low-speed interrupts, slots 8–11 high-speed. Register HL must be destroyed after you add a task. On entry to your task driver routine, the IX register contains the TCB address.

Lines 240–320 represent Listing 2's initialization routine. They set up the interrupt-handling routine in lines 400–480 and start it running. Lines 240–260 clear the

Listing 3 continued

```
00370           LD      HL,MSG1         ;GET "NOT OPERATION" MSG.
00380           CALL    DSPLY           ;DISPLAY IT
00390           RET                     ;RETURN TO CALLING PROG.
00400 MSG1      DEFM    'The interrupts have NOT been activated yet!'
00410           DEFB    0DH
00420 STRT2     LD      DE,POINT        ;LOC. OF ADRS OF PROGRAM
00430           LD      A,11            ;LAST HIGH SPEED INTERRUPT
00440           CALL    ADDTSK          ;ADD TASK TO TASK-TABLE
00450           LD      HL,0EFFFH       ;SET HI-RAM TO EFFFH.
00460           LD      (4411H),HL      ;PROTECT THIS PROGRAM
00470           RET                     ;RETURN TO CALLING PROG.
00480 ;
00490           ORG     4174H           ;HOOK FOR "CMD" COMMAND
00500           DEFW    CMDHOK          ;REPLACE WITH NEW HOOK
00510                                   ;OLD HOOK IS TO 5374H
00520 ;
00530 ; Start of Interrupt Handling Routine
00540 ;
00550           ORG     0F600H
00560 POINT     DEFW    START           ;LOCATION OF INT. ROUTINE
00570 FLAG      DEFB    0
00580 NOTIFY    DEFB    0
00590 LEN       DEFB    0               ;LOCATION OF MESSAGE
00600 OPRATE    DEFB    0
00610 LEN2      DEFB    0
00620 COUNT     DEFB    3
00630 ADDTSK    EQU     0FF52H          ;ADD A TASK TO TASK-TABLE
00640 TIMER     EQU     0FF5FH          ;33.333 MS TIMER COUNTER
00650 BUFFER    EQU     0F300H          ;BUFFER FOR MESSAGE
00660 START     LD      A,(OPRATE)      ;GET OPERATION PERMISSION
00670           CP      0               ;CAN ROUTINE OPERATE?
00680           RET     Z               ;IF NOT, RETURN FROM INT.
00690           LD      A,(16916)       ;GET SCROLL PROTECT VALUE
00700           CP      0               ;IS IT A ZERO?
00710           CALL    Z,PROTCT        ;PROTECT FIRST LINE
00720           LD      A,(COUNT)       ;GET COUNT FOR SPEED
00730           DEC     A               ;DECREMENT COUNTER
00740           LD      (COUNT),A       ;SAVE COUNT
00750           CP      0               ;IS IT TIME FOR INTRPT?
00760           RET     NZ              ;RETURN IF NOT TIME
00770           LD      A,3             ;RESET COUNTER
00780           LD      (COUNT),A       ;COUNTER RESET
00790           LD      A,(FLAG)        ;GET FLAG STATUS
00800           BIT     0,A             ;BUSY PRINTING A MSG?
00810           JP      Z,NOPRNT        ;IF NOT, RESET FLAGS
00820 PRNTNG    LD      HL,BUFFER       ;MESSAGE BUFFER AREA
00830           LD      A,(LEN2)        ;GET MESSAGE LENGTH
00840           LD      C,A
00850           LD      B,0             ;BC=MESSAGE LENGTH
00860           ADD     HL,BC           ;POINT TO CHAR. TO PRINT
00870           INC     A
00880           LD      (LEN2),A
00890           LD      C,A
00900           LD      A,(LEN)
00910           CP      C
00920           CALL    Z,RESFLG        ;QUEUE NOW AVAILABLE
00930           LD      (LEN),A
00940           LD      A,(HL)          ;GET CHARACTER TO PRINT
00950           LD      HL,SCREEN+1
00960           LD      DE,SCREEN
00970           LD      BC,63           ;63 CHARACTERS TO MOVE
00980           LDIR                    ;SCROLL THEN
00990           LD      (SCREEN+63),A   ;SAVE NEW CHARACTER
01000           LD      A,(FLAG)        ;GET FLAG STATUS
01010           BIT     0,A             ;LAST CHAR. PRINTED?
01020           RET     NZ              ;RETURN IF NOT
01030           LD      A,(NOTIFY)      ;FINISHED CLEARING SCREEN
01040           CP      2
01050           JR      Z,CLRFLG        ;IF SO, RESET BIT 1
01060           CALL    ADJUST          ;ADJUST TO CLEAR SCREEN
01070           RET                     ;RETURN FROM INTERRUPT
01080 CLRFLG    LD      A,(LEN2)        ;GET CHAR. COUNT
01090           LD      C,A             ;SAVE IT
01100           LD      A,(LEN)         ;GET MESSAGE LENGTH
01110           CP      C               ;DONE PRINTING IT?
01120           RET     NZ              ;IF NOT, RETURN TILL DONE
01130           LD      A,(FLAG)        ;GET FLAG STATUS
01140           RES     1,A             ;RESET CLEAR SCREEN FLAG
01150           LD      (FLAG),A        ;FLAG RESET
01160           XOR     A               ;ZERO A REGISTER TO
01170           LD      (LEN),A         ;CLEAR THIS FLAG
01180           LD      (LEN2),A        ;AND THIS FLAG
01190           LD      (NOTIFY),A      ;... AND THIS FLAG
01200           RET                     ;RETURN FROM INTERRUPT
01210 NOPRNT    LD      A,(FLAG)        ;GET FLAG STATUS
01220           BIT     1,A             ;BUSY, BUT AVAILABLE?
01230           JR      NZ,PRNTNG       ;CONTINUE PRINTING
01240           BIT     2,A             ;ANOTHER WAITING QUEUE?
01250           JR      Z,NONEW         ;IF NOT, RESET FLAGS
01260           RES     2,A             ;RESET WAITING QUEUE
01270           RES     1,A             ;RESET CLEARING FLAG
01280           SET     0,A             ;BUSY PRINTING A MESAGE
01290           LD      (FLAG),A        ;SAVE FLAG STATUS
01300           XOR     A
01310           LD      (NOTIFY),A
01320           JR      PRNTNG          ;BEGIN PRINTING MESSAGE
01330 NONEW     LD      A,0             ;THE QUEUE IS EMPTY
01340           LD      (NOTIFY),A      ;THE QUEUE IS AVAILABLE
01350 RESFLG    PUSH    AF              ;SAVE AF REGISTER
01360           LD      A,(FLAG)        ;GET FLAG STATUS
01370           RES     0,A             ;NOT BUSY PRINTING
01380           RES     1,A             ;NOT BUSY PRINTING
01390           LD      (FLAG),A        ;SAVE FLAG STATUS
01400           XOR     A
```

Listing 3 continued

Listing 3 continued

```
01410           LD      (LEN),A         ;RESET FLAGS
01420           LD      (LEN2),A        ;RESET FLAGS
01430           POP     AF              ;RESTORE AF
01440           RET                     ;RETURN
01450 ADJUST    PUSH    HL              ;SAVE REGISTERS
01460           PUSH    AF
01470           LD      A,64            ;GET LENGTH OF MESSAGE
01480           LD      (LEN),A         ;ADJUST POR SCREEN CLEAR
01490           LD      HL,BUFFER
01500           LD      C,A
01510           XOR     A               ;ZERO A REGISTER
01520           LD      (LEN2),A        ;RESET THIS COUNTER
01530 LOOP1     LD      (HL),20H        ;FILL-IN WITH A SPACE
01540           INC     HL
01550           DEC     C               ;BUMP POINTERS & COUNTERS
01560           LD      A,C
01570           CP      0               ;COUNTER = 0 ?
01580           JR      NZ,LOOP1        ;CONTINUE UNTIL DONE
01590           LD      A,(FLAG)        ;GET FLAG STATUS
01600           SET     1,A             ;BUSY, BUT AVAILABLE
01610           RES     0,A             ;QUEUE IS AVAILABLE
01620           LD      (FLAG),A        ;SAVE FLAG STATUS
01630           LD      A,2             ;BUSY, BUT AVAILABLE
01640           LD      (NOTIFY),A      ;NOTIFY BASIC OF THIS
01650           POP     AF              ;RESTORE REGISTERS
01660           POP     HL
01670           RET                     ;RETURN
01680 PROTCT    LD      A,1             ;SCROLL PROTECT 1 LINE
01690           LD      (16916),A       ;PROTECT IT
01700           RET                     ;RETURN TO PROGRAM
01710 ;
01720 ; Start of hook routine to print things using interrupt
01730 ;
01740 CNDHOK    PUSH    AF              ;SAVE CONDITION CODES
01750           PUSH    DE
01760           EI                      ;ENABLE INTERRUPTS
01770           LD      A,(HL)          ;GET COMMAND SYNTAX
01780           CP      '!'             ;IS IT A VALID COMMAND?
01790           JR      Z,GOOD          ;IF SO, EXEC NEW ROUTINE
01800           POP     DE              ;RESTORE REGISTERS
01810           POP     AF              ;RESTORE CONDITION CODES
01820           JP      5374H           ;EXECUTE THE BASIC COMAND
01830 GOOD      LD      A,1             ;TURN ON THE INTERRUPT
01840           LD      (OPRATE),A      ;SUBROUTINE
01850 GOOD2     LD      A,(FLAG)        ;GET FLAG STATUS
01860           BIT     0,A             ;QUEUE AVAILABLE?
```

Listing 3 continued

screen and notify you that the interrupt is working. Lines 300–310 protect the interrupt from other data loading in memory.

Line 270 loads the TCB address ("Main") into DE. Line 390 shows you that Main points to Main2, the task driver's entry point.

Line 280 loads A with the task's slot assignment: You have 12 slots numbered zero to 11. In this case, the slot is zero, a low-speed interrupt. Line 290 calls the routine that adds the interrupt to the task table, which contains the 12 interrupts' TCBs. Don't fool around with these locations in memory or the program might crash. Finally, line 320 exits to TRSDOS Ready, marking the end of the initialization procedure.

Lines 400–480 make up the task driver routine. Note that when the task has executed, it must return from the interrupt. Don't ever use a jump instruction to exit the routine or your computer will bomb. Lines 450 and 480 contain the return instructions, which return the processor from the interrupt so that the program it interrupted can continue running.

To remove a task from the task table, all you have to do is specify in the A register which slot contains the task you want to remove, and call the REMTSK subroutine to do so. HL and DE are destroyed after the call to this subroutine.

Listing 3 continued

```
01870           JR      NZ,GOOD2        ;IF NOT, WAIT UNTIL IT IS
01880           CALL    BUTTIN          ;ABORT PRESENT DUTY
01890           LD      C,0             ;ZERO THE COUNTER
01900           INC     HL              ;POINT TO DATA TO PRINT
01910           INC     HL              ;BYPASS THE '"' SYMBOL
01920           LD      DE,BUFFER       ;POINT TO BUFFER AREA
01930 LOOP2     LD      A,(HL)          ;GET A DATA BYTE TO PRINT
01940           LD      (DE),A          ;STORE CHAR IN BUFFER
01950           INC     HL              ;BUMP POINTERS
01960           INC     DE
01970           INC     C
01980           CP      0               ;LAST CHARACTER STORED?
01990           JR      Z,LOOP3         ;IF SO, THEN FINISHED
02000           CP      '"'             ;TERMINATER?
02010           JR      Z,ADJLDP        ;IF SO, ADJUST HL POINTER
02020           JR      LOOP2           ;CONTINUE UNTIL DONE
02030 LOOP3     LD      A,C             ;AND STORE THE COUNTER
02040           LD      (LEN),A         ;VALUE IN STORAGE SLOT
02050           LD      A,(FLAG)        ;GET FLAG STATUS
02060           SET     0,A             ;WAITING FOR THE QUEUE
02070           RES     1,A
02080           RES     2,A
02090           LD      (FLAG),A        ;SAVE FLAG STATUS
02100           DEC     HL              ;CORRECT POINTER
02110           POP     DE              ;RESTORE REGISTERS
02120           POP     AF              ;RESTORE CONDITION CODES
02130           RET                     ;CONTINUE ON WITH PROGRAM
02140 BUTTIN    LD      A,(FLAG)        ;GET FLAG STATUS
02150           RES     0,A             ;NOT BUSY...
02160           RES     1,A             ;NOT BUSY NOW...
02170           LD      (FLAG),A        ;SAVE FLAG STATUS
02180           XOR     A               ;ZERO A TO RESET
02190           LD      (LEN),A         ;THIS FLAG...
02200           LD      (LEN2),A        ;AND THIS FLAG...
02210           LD      (NOTIFY),A      ;AND THIS FLAG
02220           RET                     ;RETURN TO CALLER
02230 ADJLDP    DEC     DE              ;POINT TO THE '"' SIGN
02240           XOR     A
02250           LD      (DE),A          ;ZERO IT OUT
02260           INC     DE              ;REPOSITION DE
02270 ADJLP2    LD      A,(HL)          ;CONTINUE UNTIL FOUND 0
02280           INC     HL              ;BUMP POINTER
02290           CP      0               ;END OF COMMAND LINE?
02300           JR      NZ,ADJLP2       ;CONTINUE UNTIL DONE
02310           JR      LOOP3           ;FINISHED LOOP
02320           END     START
```

End

Take Command

Program Listing 3, Scroll, is a good example of a task driver's power. It adds a command to Basic, CMD!, that scrolls characters across the top of the screen. Table 2 shows Scroll's important addresses.

To install Scroll, assemble Listing 3 to disk and make sure you've enabled Break In. Now go into Basic and type in:

```
CMD"L","SCROLL/CMD":DEFUSR=&HF200:
A=USR(0)
```

This loads Scroll into memory and runs it. If you haven't enabled Break In, an error message appears and Scroll aborts.

Try out the new command by typing in:

```
CMD!"ABLE WAS I ERE I SAW ELBA."
```

You should see the message scroll across the top of your screen. The program scroll-protects the screen's top line; to unprotect it, you have to turn off the Scroll function by typing in:

```
POKE &HF005,0
```

To restore scroll protection, POKE the same location with a value other than zero or invoke the CMD! command again.

You can change the scrolling speed. For faster scrolling, POKE locations F007 and F021 with values less than 3 but not zero. The value 3 represents the original scrolling speed. To slow down the scroll, POKE the locations with a value greater than 3,

Address (hex)	Description
F200	Start of Scroll's initialization procedure.
F000	A pointer Break In uses to find the task routine's entry point.
F002	A flag byte indicating Scroll's status.
F004	Contains the message's length.
F005	Operation permission byte. If this is anything other than 1, the program is off. If it's 1, the program is on.
4174	Hook location for Basic's new CMD! command.
F007 and F021	Changing these values speeds up or slows down scrolling.
F300–FFFF	Message buffer area.

Table 2. Scroll's routines.

but less than 256, or zero. If you decide to change speeds, be sure to POKE the same number in both memory locations.

If you type in two messages to scroll, the computer waits until the first finishes scrolling before printing the second.

Location F004 contains the length of the message being printed, which can be up to 256 characters. Location F006 contains the number of characters printed so far. When F006 equals F004, the message has finished printing.

Location F002 is a flag value containing Scroll's status. Here's a rundown on the bits in this byte:

Bit zero	If set, the program is printing a message and the queue is unavailable.
Bit 1	If set, the program is scrolling a message off the screen, but is available to print another message.
Bit 2	If set, another message is waiting in the queue. When the program finishes scrolling the first message, it prints the message in the queue and resets this bit.
Bits 3–7	Scroll doesn't use these bits, so they're available for your use.

Don't touch bits zero to 2 of this byte or you could really mess things up.

Scroll keeps the message it's printing in a buffer at location F300 and reserves 256 bytes for the buffer area. The scroll interrupt occupies slot 11 of the task table, so don't use this slot for another interrupt if you want Scroll to operate concurrently.

RAMifications

To run the Listings on a Model III with less than 48K of RAM, change their ORG addresses to appropriate values. But make sure that no other task drivers load over Listing 1 while it's running.

You should originate Listing 1 so that the program's last byte loads into the highest possible RAM location—this gives you the maximum amount of free memory. If you do change the programs' loading addresses, the POKEs and other addresses described above won't apply.■

Cary Oler has been working with computers for five years. You can write to him at Box 132, Stirling, Alberta, Canada TOK, 2EO.

Related Articles

Fisher, Douglas C., "Interrupt Your 80," January 1983, p. 258. Maskable and nonmaskable interrupts for the Model I.

Genovese, R.F., "Multi-Programming on a Micro," January 1982, p. 278. A Model I interrupt program.

Garsky, Buzz, "Doing Two Things at Once," March 1981, p. 178. A Model I tutorial on interrupts.

Workman, Dennis, "We Interrupt This Program," November 1982, p. 396. Using interrupts to speed up the Model I.

The Right Address

Different versions of TRSDOS 6.X use different system memory addresses. Locator gives you the right addresses for Model 4 TRSDOS.

Finding a memory address in Model 4 TRSDOS is a bit like finding Main St. in five different towns; the idea is the same in each, but the location changes. So it is with memory addresses under TRSDOS 6.X—they change with each version of TRSDOS.

Locator (see Program Listing 1) determines the correct memory addresses for cursor control, scroll protection, redefinable function keys, and the keyboard for your version of TRSDOS. It does so by searching low memory for the location of $DO and $KI and adding displacements to these locations to get the right addresses.

You can store these addresses on disk so Basic programs can access them. Since the memory locations on disk are correct for the resident version of TRSDOS, Basic programs operating under control of the system disk will automatically use the proper PEEK and POKE addresses.

Variable	Function
LS	Scroll protection
LC	Cursor image
L1	Function key 1, lowercase
L2	Function key 1, uppercase
L3	Function key 2, lowercase
L4	Function key 2, uppercase
L5	Function key 3, lowercase
L6	Function key 3, uppercase
LK	Keyboard map

Table. Locator's subroutine saves memory addresses to these variables.

To further facilitate programming, I've included a Basic subroutine (see Program Listing 2) that copies the addresses on the system disk to the program variables indicated in the Table. To accomplish this, you execute a GOSUB command to the subroutine.■

You can write to Maurice Dyke at P.O. Box 32077, Aurora, CO 80041.

Program Listing 1. Locator.

```
10   'LOCATOR---BASIC PROGRAM BY M. DYKE
20   N = 0: JK = 0: JV = 0
30   FOR J = 1 TO 5000
40   JP = PEEK(J)
50   IF N <> 0 THEN 150
60   IF PEEK(J) <> 84 THEN 150
70   IF PEEK(J+1) <> 82 THEN 280
80   IF PEEK(J+2) <> 83 THEN 280
90   IF PEEK(J+3) <> 68 THEN 280
100  IF PEEK(J+4) <> 79 THEN 280
110  IF PEEK(J+5) <> 83 THEN 280
120  IF PEEK(J+6) <> 54 THEN 280
130  N = PEEK(J+7)
140  GOTO 280
150  IF JP <> 36 THEN 280
160  J1 = PEEK(J+1)
170  IF JK <> 0 THEN 230
180  IF J1 <> 75 THEN 230
190  IF PEEK(J+2) <> 73 THEN 280
200  JK = J
210  IF JV <> 0 THEN 330
220  GOTO 280
230  IF JV <> 0 THEN 280
240  IF J1 <> 68 THEN 280
250  IF PEEK(J+2) <> 79 THEN 280
260  JV = J
270  IF JK <> 0 THEN 330
280  PRINT "CHECKING LOC ",J
290  NEXT J
300  CLS
310  PRINT "MEMORY SEARCH NOT SUCCESSFUL FOR OPERATING SYSTEM IN USE"
320  GOTO 520
330  CLS
340  PRINT "SOME USEFUL MEMORY LOCATIONS IN TRSDOS 6.";CHR$(N)
350  PRINT " "
360  PRINT " SCROLL PROTECTION: ";JV+7;"  ( POKES 9-15 )"
370  PRINT "  CURSOR CHARACTER: ";JV+11
380  PRINT "    LC F1 CHARACTER: ";JK+35
390  PRINT "    UC F1 CHARACTER: ";JK+36
400  PRINT "    LC F2 CHARACTER: ";JK+37
410  PRINT "    UC F2 CHARACTER: ";JK+38
420  PRINT "    LC F3 CHARACTER: ";JK+39
430  PRINT "    UC F3 CHARACTER: ";JK+40
440  PRINT "KEYBOARD MAP START: ";JK+11
450  PRINT "  KEYBOARD MAP END: ";JK+18
460  LINE INPUT"ENTER Y TO SAVE ON DISK FOR USE BY OTHER BASIC PROGRAMS ";Y$
470  IF Y$ <> "Y" THEN 520
480  OPEN "O",1,"SYSLOC/TXT:0"
490  WRITE#1, JV+7,JV+11,JK+35,JK+36,JK+37,JK+38,JK+39,JK+40,JK+11
500  CLOSE 1
510  PRINT"DATA STORED IN FILE 'SYSLOC' FOR USE BY OTHER BASIC PROGRAMS"
520  END
```

End

Program Listing 2. Basic subroutine to copy variables.

```
65000 'PROGRAM FILE "GETLOC" A SUBROUTINE BY M. DYKE FOR INCLUSION IN OTHER BASI
C PROGRAMS TO GET MEMORY LOCATIONS STORED ON SYSTEM DISC BY PROGRAM "LOCATOR"
65010  'LC=CURSOR L5=SCROLL PROTECTION L1=F1LC L2=F1UC L3=F2LC L4=F2UC L5=F3LC
       L6=F3UC LK=KEYBOARD MAP
65020  ON ERROR GOTO 65030:  OPEN "I", 1, "SYSLOC/TXT:0": ON ERROR GOTO 0: INPUT #
1, LS,LC,L1,L2,L3,L4,L5,L6,LK: CLOSE 1: RETURN
65030  PRINT"PROGRAM 'LOCATOR' MUST BE EXECUTED BEFORE CURRENT PROGRAM CAN BE RUN
": RESUME 65040
65040  END
```

End

LOAD 80

System Requirements

Model 4/4P
32K RAM
TRSDOS 6.X

Rembrandt Redux

Our hi-res MacPaint-like program revisited—with screen dump routines for Epson printers and some TRSDOS 1.3 patches to BasicG.

I like 80 *Micro*'s high-resolution MacPaint-style graphics program, Rembrandt, ("Drawing in Detail," September 1985, p. 56), but Model III owners can't execute BasicG's GSAVE, GLOAD, or GPRINT commands from within Rembrandt; it returns control to TRSDOS 1.3 if you do so. In addition, you can't use Rembrandt with an Epson printer. I'll show you how you can do both.

To fix Rembrandt for operation under TRSDOS 1.3, add these lines to the program:

```
1 GOTO 5
2 FOR I=1 TO LEN(FI$):POKE-249+I-1,ASC
  (MID$(FI$,I,1)):NEXT:X=USR(0):RETURN
5 CLEAR500:FOR I=0 to 6:READ A:PDKE
  &HFF00+I,A :NEXT :DEFUSR=&HF00
  :DATA 33, 7, 255, 205, 156, 66, 201
```

Also, you have to change some Rembrandt lines to those in the Figure. This adds a small machine-language program that uses the CMDDOS call (429CH) to execute a TRSDOS command. You should set memory size to 61439 (0F000H) since this is where the graphics routines load.

The screen print routines that come with BasicG don't work with Epson printers. I modified the routines in the BasicG manual to work on the Epson MX-80 and FX-80. Program Listing 1 prints the screen with the X axis down the page and the Y axis across it; i.e., I rotated the screen 90 degrees. The routine prints the dots on the Y axis twice.

Program Listing 2 prints the *X* axis across the page and the Y axis down it. Neither of these routines use BasicG's screen, ROM, or supervisor calls.

The routine to initialize the Model III graphics board is in the *TRS-80 Computer Graphics Operation Manual*, Radio Shack catalog #26-1125, pp. 89 and 90. Insert lines 125–154 from the manual where indicated in Listings 1 and 2.■

You can write to Dale Elton Rogerson at 1123B McMillian St., Atlanta, GA 30332.

LOAD 80

System Requirements

Model III
64K RAM
BasicG
High-resolution board

Program Listing 1. Epson screen dump routine that prints the X axis down a page and the Y axis across. N.B.: You must insert several lines where indicated from an initialization routine in your BasicG manual.

```
                00000
                00001 ;************************************
                00002 ;*        EPSON SCREEN DUMP 1       *
                00003 ;*                by                *
                00004 ;*           Dale Rogerson          *
                00005 ;*            March 84              *
                00006 ;*  For Hi-Res Board (III)          *
                00007 ;* Complete re-write of GPRINT.      *
                00008 ;* Dumps screen to an Epson printer: *
                00009 ;* FX-80,MX-80,RX-80 or compatible.  *
                00010 ;* Prints Y axis across page with    *
                00011 ;* each screen line printed twice.   *
                00012 ;* This dump fills a whole page.     *
                00013 ;************************************
                00014 ;
F000            00015          ORG     0F000H
F000 E5         00016 GPRINT   PUSH    HL              ;Save the Regs
F001 D5         00017          PUSH    DE
F002 C5         00018          PUSH    BC
F003 DDE5       00019          PUSH    IX
F005 CD0000     00020          CALL    INITG           ;Initialize Graphics
F008 3E51       00021          LD      A,81    ;81=1000001B Inc X on Read & write
F00A D383       00022          OUT     (STATUS),A      ;Set Status
F00C 2195F0     00023 SETUP    LD      HL,NUMPIN       ;Set Printer for 8 pins
F00F 0603       00024          LD      B,3             ;3 bytes
F011 CD45F0     00025 SETUP2   CALL    PRINTA          ;Print byte
F014 23         00026          INC     HL              ;Get next byte
F015 10FA       00027          DJNZ    SETUP2          ;Go print again
F017 0650       00028 FORX     LD      B,80    ;B=number of columns to Print
F019 2192F0     00029          LD      HL,BUFFER;NL==> Buffer
F01C 78         00030 FORX2    LD      A,B     ;A=B
F01D 3D         00031          DEC     A       ;Column # = B-1
F01E D388       00032          OUT     (X),A   ;Set X position
F020 AF         00033          XOR     A       ;A=0
F021 4F         00034          LD      C,A     ;C= line # (screen)
F022 D381       00035          OUT     (Y),A   ;Set Y position to 0
F024 C5         00036          PUSH    BC      ;Save # of columns
F025 CD4FF0     00037          CALL    GRAMOD  ;Put printer in Graphics mode
F028 DB02       00038 FORY     IN      A,(GRAPH)       ;Get Byte
F02A CD5DF0     00039          CALL    REVERS          ;Byte backwards-Reverse
F02D 77         00040          LD      (HL),A          ;Put Byte into BL
F02E CD45F0     00041          CALL    PRINTA          ;Print Byte
F031 CD45F0     00042          CALL    PRINTA          ;Print Byte again
F034 0C         00043          INC     C               ;Inc Line #
F035 3EF0       00044          LD      A,240           ;A=last screen line #
F037 B9         00045          CP      C               ;At last screen line?
F038 20EE       00046          JR      NZ,FORY         ;If not print next byte
F03A 360A       00047          LD      (HL),0AH        ;Print a line feed
F03C CD45F0     00048          CALL    PRINTA
F03F C1         00049          POP     BC      ;Get counter
F040 10DA       00050          DJNZ    FORX2   ;Do next printer line
F042 C378F0     00051          JP      BYE     ;Finished so go end it
                00052 ;-------Print a Byte
F045 DBFB       00053 PRINTA   IN      A,(251)         ;Check Printer Status
F047 FE3D       00054          CP      61              ;Ready?
F049 20FA       00055          JR      NZ,PRINTA       ;Check again if not
F04B 7E         00056          LD      A,(HL)          ;Print Byte
F04C D3FB       00057          OUT     (251),A
F04E C9         00058          RET                     ;Return
                00059 ;--------Put Printer in Graphics Mode
F04F E5         00060 GRAMOD   PUSH    HL
F050 218EF0     00061          LD      HL,BGMODE
F053 0604       00062          LD      B,4
F055 CD45F0     00063 GRA001   CALL    PRINTA
F058 23         00064          INC     HL
F059 10FA       00065          DJNZ    GRA001
F05B E1         00066          POP     HL
F05C C9         00067          RET
                00068 ;-------Reverse the Byte in A
F05D 3298F0     00069 REVERS   LD      (XLOC),A        ;Save the byte
F060 AF         00070          XOR     A               ;ZERO A
F061 0601       00071          LD      B,1
F063 118000     00072          LD      DE,80H  ;D = New Byte/E = Mask Byte
F066 3A98F0     00073 START    LD      A,(XLOC);Get byte back
F069 A3         00074          AND     E       ;Use mask to get bit
```

Listing 1 continued

PRINTER SALE

THE STATE OF THE ART

The C.ITOH 3500 is fully IBM/Tandy 1000-1200-2000 compatible and FAST. Using bidirectional, logic-seeking printing across a full width of 13.2 inches, this little beauty prints at a speed of 350 characters per second (CPS). Letter quality printing is available at a rapid *87* CPS—more than twice the speed of most daisywheel printers. Couple that with a quick linefeed of 30 milliseconds per line and a standard buffer of 2K (16K optional, $50) and you have just about the fastest throughput around. Only 4.7" high and e low noise level of 58 dBA.

Maximum versatility is offered for data and wordprocessing output, spreadsheets and business graphics by a variability of print densities, speeds, character sets and fonts, spacing, forms control and other attributes realized by operator and computer alterable functions, allowing the printer to be tailored to almost any application.

We could rave on and on about this fantastic printer but the proof is using this printer in your application. Try it on our 14 day money-back-if-not-satisfied plan. Don't be printer bound any longer. Call today.

350 CPS only $1695
(Please Specify Serial or Parallel)
(Shipping Included)

C.ITOH DIGITAL PRODUCTS

Low Profile Design
ONE YEAR WARRANTY

$229 (Add $10 S & H)
List $299

ProWriter™ Jr. Near letter quality and 105 CPS for data processing. Compatible with all popular PCs. Built-in stand, easy front paper loading. The low cost answer for professional performance.

StarWriter™ Y-10, F-10 40/55. Daisy wheels with speeds ranging from 22 to 58 CPS for the sharpest in letter quality printing. 13.5 and 15-inch carriage widths. Switch-selectable pitches 10,12, and 15. Compatible with all popular word processing software.

As Low As **$899**
(Add $15 S & H)

F10-40 15" wide 40cps Diablo/Qume printwheels....$899
F10-55 Fastest DW printer at 58cps 15" wide $999
F10 Bi-Directional Tractor $199
F10 Mechanical sheet feeder $299

As Low As **$299**
(Add $12 S & H)

ProWriter™ 8510S/1550S + NLQ Series Dot Matrix Printers. Near letter quality, 180/120 CPS with throughput speeds of up 110 LPM for far faster text and graphics than competitors. Models specifically designed to provide the finest performance and flexibility with IBM® and Apple®/Macintosh® microcomputers.

8510 BPI IBM Graphics, 120cps 10" friction/tractor 5299
8510 SEP IBM Graphics, NLQ and faster at 180cps $399
8510 CEP IBM Graphics, 180cps plus 7 colors $499
1550 EP 15" wide 120cps friction/tractor included 5499
1550 BCD Like above but *RS232 interface* .. **SPECIAL....$299**
1550 SEP 15" IBM Graphics NLQ and a fast 180cps $599

ProWriter™ 24LQ. The Ultimate Printer. 24-pin dot matrix printer. 6 printers in 1 for all professional printing needs. 200 CPS for data, 133 CPS memo quality, 67 CPS letter quality. Selectable type styles. Up to 7-color graphics at 360 × 360 DPI.

$999 (Add $15 S & H)
List $1299

ORDER TODAY
CALL TOLL FREE **800-527-0347/800-442-1310**

MONTEZUMA MICRO

P.O. Box 32027
Redbird Airport, Hangar #8
Dallas, TX 75232
214-339-5104

MasterCard VISA AMERICAN EXPRESS

Listing 1 continued

```
F06A EA70F0   00075          JP      PE,NXTONE;Go if bit not set
F06D 78       00076          LD      A,B       ;Get Mask Byte
F06E B2       00077          OR      D         ;Merge with New Byte
F06F 57       00078          LD      D,A       ;Put New byte into D
F070 CB0B     00079 NXTONE   RRC     E         ;Nask next bit
F072 CB00     00080          RLC     B         ;Done all bits and back to 7?
F074 30F0     00081          JR      NC,START;Go to start if not all done
F076 7A       00082          LD      A,D       ;Put new byte into A
F077 C9       00083          RET               ;Return
              00084 ;--------Return Printer to Normal & End Program
F078 2193F0   00085 BYE      LD      HL,EGMODE       ;Set printer to Normal
F07B 0602     00086          LD      B,2
F07D CD45F0   00087 BYE2     CALL    PRINTA
F080 23       00088          INC     HL
F081 10FA     00089          DJNZ    BYE2
F083 3EFC     00090          LD      A,0FCH  ;Set Options
F085 D383     00091          OUT     (STATUS),A
F087 DDE1     00092          POP     IX      ;Get Regs
F089 C1       00093          POP     BC
F08A D1       00094          POP     DE
F08B E1       00095          POP     HL
F08C AF       00096          XOR     A
F08D C9       00097          RET
              00098 ;-------Data
0080          00099 X        EQU     80H
0081          00100 Y        EQU     81H
0082          00101 GRAPH    EQU     82H
0083          00102 STATUS   EQU     83H
F08E 1B       00103 BGMODE   DEFB    27      ;Graphics mode
F08F 4B       00104          DEFB    'K'
F090 E0       00105          DEFB    224
F091 01       00106          DEFB    1
0001          00107 BUFFER   DEFS    1
F093 1B       00108 EGMODE   DEFB    27      ;Normal Mode
F094 40       00109          DEFB    '@'
F095 1B       00110 NUMPIN   DEFB    27      ;Set number of Pins
F096 41       00111          DEFB    'A'
F097 08       00112          DEFB    8
F098 00       00113 XLOC     DEFB    0
              00114 ;-------Initialize Graphics Board-Found in Manual
              00115 ; Insert lines 125-154 of the Initialization routine
              00116 ; page 89-90 of the TRS-80 Computer Graphics Operation
              00117 ; Manual.  Radio Shack Catalog # 26-1125.
F000          00155          END     GPRINT
```

End

Program Listing 2. Epson screen dump that prints the X axis across page and the Y axis down. N.B.: You must insert several lines where indicated from an initialization rautine in your BasicG manual.

```
              00000
              00001 ;************************************
              00002 ;*          EPSON SCREEN DUMP 2       *
              00003 ;*                by                  *
              00004 ;*            Dale Rogerson           *
              00005 ;*              May 85                *
              00006 ;*    For Hi-Res Graphics board (III)*
              00007 ;* Complete re-write of GPRINT.       *
              00008 ;* Dumps screen to an Epson Printer   *
              00009 ;* FX-80,MX-80,RX-80 or Compatible.   *
              00010 ;* Prints X axis across page.         *
              00011 ;************************************
              00012 ;
F000          00013          ORG     0F000H
F000 E5       00014 GPRINT   PUSH    HL              ;Save registers
F001 D5       00015          PUSH    DE
F002 C5       00016          PUSH    BC
F003 DDE5     00017          PUSH    IX
F005 CD0000   00018          CALL    INITG           ;Initialize Graphics
F008 3ED1     00019          LD      A,209   ;11010001B - inc y on read
F00A D383     00020          OUT     (STATUS),A      ;Set options
F00C 210DF1   00021 SETUP    LD      HL,NUMPIN       ;HL==> ESC'A8'
F00F 0603     00022          LD      B,3             ;Sets # of pins to 8
F011 CDD9F0   00023 SETUP2   CALL    PRINTA          ;Send to printer
F014 23       00024          INC     HL
F015 10FA     00025          DJNZ    SETUP2
F017 DD2112F1 00026          LD      IX,SCRBUF       ;IX = 8 byte buffer
F01B 0E82     00027          LD      C,82H           ;PORT
F01D AF       00028          XOR     A               ;Zero A
F01E 3211F1   00029          LD      (POSY),A        ;Set Y position to 0
F021 061E     00030          LD      B,30            ;NUMBER OF PRINTER LINES
F023 C5       00031 OUT      PUSH    BC              ;SAVE NUMBER
F024 CDE3F0   00032          CALL    GRAMOD          ;Printer in Graphics mode
F027 211AF1   00033          LD      HL,PRTBUF       ;HL
F02A AF       00034          XOR     A               ;ZERO X POSITION
F02B 3210F1   00035          LD      (POSX),A        ;SAVE IT
F02E 0650     00036          LD      B,80            ;B=# of Columns to Print
F030 C5       00037 MIDDLE   PUSH    BC              ;Save count
F031 3A10F1   00038          LD      A,(POSX)        ;Get X-position
F034 D380     00039          OUT     (80H),A         ;Set it
F036 3A11F1   00040          LD      A,(POSY)        ;Get Y-position
F039 D381     00041          OUT     (81H),A         ;& set it
F03B ED78     00042          IN      A,(C)           ;Get byte at screen loc
```

Listing 2 continued

Circle 347 on Reader Service card.

TRS 80 Computers

All Tandy Computer Products Available
Manufacturer's Warranty

Model 1000
Model 1200 HD
Model 2000
Model 4
Model 100/200
26-5111 Monitor

New Tandy Printers Available

Unbeatable Year End Clearance Prices

Visa — MasterCard — American Express
Cashier's Check — Money Order

Business Telephone Systems—Discount Prices—
Install your own. Completely modular.
Call for Prices

Computer Specialist available for assistance.

TALLEY COMMUNICATIONS CO.

P.O. Box 193 • 121 N. State St.
Decatur, Texas 76234 • 817-627-2553

Call for other computer, telephone and accessory prices!

Fast Delivery — References Available

Circle 46 on Reader Service card.

Hypercross gets better and better!

*** TRS80 - CP/M - MS-DOS - CoCo File Transfer ***

Now you can CROSS the barrier between computers! Using **HYPERCROSS** you can **COPY** files between TRS-80 disks and those from many **CP/M** and **IBM-PC** type computers. If you have access to more than one kind of computer, or you are changing to a new machine then you need HYPERCROSS to transfer your text files, BASIC, FORTRAN PASCAL or C programs, Viscalc files, general ledger and accounting files, data bases and even binary files. HYPERCROSS lets you format alien disks and copy files on your own **TRS-80 or MAX-80** without using cables.

Formats supported: IBM-PC and MS-DOS compatibles include DOS 1.1, 2.x/3.0 single and double sided and Tandy 2000 CP/M from Aardvark to Zorba, including all pppular TRS80 CP/M formats.

PRICES Inc disk manuel, S/H. We will match any advertised price.

Hypercross CoCo with TRS80-Color Computer . . . . . NEW! . . . . **$49.95 ppd**
Hypercross CP/M with 40 single sided formats . . . . . **$49.95 ppd**
Hypercross PC/MS-DOS standard formats . . . . . **$49.95 ppd**
Hypercross XT/2.0 with 90 CP/M and PC formats . . . . Special! . . . **$93.95 ppd**
Hypercross XT/2.0-Plus. Now with 200+ formats inc CoCo . . . . . **$139.95 ppd**

Upgrade at any time for price difference plus $5 plus old disk.
Please specify TRS-80 Model I (needs doubler), III, 4/4P, or MAX-80.

HYPERZAP 3.2! Disk Magic!

Do you want to back up your precious copy of Copycat 3, or SU. Do you want to fix or modify a disk - if so then **you need HYPERZAP!** On the market for 3 years, **HYPERZAP** is **more** than just another disk copying program - it is **the** program for analyzing, copying, repairing, creating floppy disks of all kinds. It works with TRS-80 formats as well as many others such as CP/M, PC, CoCo etc. Designed to handle mixed density sectors on any track in any sequence. Many features for reading, writing, editing track and sector data. **Hyperzap** is **the** tool that lets you be in charge. **Make your own self booting disks.** Take your own CMD file and turn it into a duel booting Mod 1/III/IV disk. **Autopilot mode** allows learns, saves and repeats procedures. Disk comes with fascinating examples. Use Hyperzap as a learning tool - find out how things are done!

HYPERZAP for Model 1/3/4 or MAX-80 (specify) . . . . . **$49.95 ppd**

Arranger II Disk Index System

World's finest disk cataloging system, Now you can find that file when you want it. Arranger will CATALOG, SORT and FIND up to 11000 files fast! Runs on any Model I, III or IV and automatically recognizes any DOS even double sided.

Arranger II - highly recommended . . . . . **$52.95 ppd**

New for 128K Model 4 users!

TMDD The Memory Disk Drive Allows New DOS-80 users to use the extra 64K memory bank as an instant electronic disk drive. Use it as your system drive! Works with Hypercross, no need for DOS in drive 0.

TMDD The Memory Disk Drive for Model 4/4P with 128K . . . . . **#31.95 ppd**

ZIPLOAD - a real time saver! fed up with waiting for MODELA to load or setting up DOS and MEMDISK? Now ZIPLOAD lets you make yourself a super fast loading self booting disk for rapid loading of DOS, Memory disk and ROM image. Example: loads Model 3 DOS and ROM image in 1.4 sec after reset delay. Works with Model 3 and 4 DOSs.

ZIPLOAD great for Model 4 and 4P . . . . . **$31.95 ppd**

HYPERSOFT
PO Box 5115S, Raleigh, NC 27609
(919) 847-4779 6-11 pm EST
Check, COD, Mastercard and Visa Accepted.

Circle 534 on Reader Service card.

Mail-Order Electronics — *Seasons Greetings!* — **Worldwide Since 1974**

TTXpress

Size: 11.2"L x 4.6"W x 1.8"H • Weight: 2.2 lbs.

Holiday SALE!

FULL FUNCTION PORTABLE PRINTER

Part No.	Description	Price
TTX Printer (Model 1280)	Includes *TTXpress* Printer, Thermal Paper Roll, Paper Roll Holder, 4 C-Size Alkaline Batteries, FREE Interface Cable (compatible with TRS-80 Model 100, Tandy Model 200 and NEC-PC8201A/8401A) and Manual.	$99.95
Accessory Kit 1280	Includes Carrying Case, AC Adapter, Battery Protector one roll of Thermal Paper.	$29.95
Thermal Paper	100 Sheets of 8½" x 11" Thermal Paper.	$ 5.95

• TRS-80 MODEL 100 • NEC • OLIVETTI

Easy to install module plugs right into the socket increasing memory in 8K increments. Three modules will increase your memory to its full capacity. Complete with module and documentation for installation.

M1008K (TRS-80 Model 100 Expansion). . . . . **$29.95 ea. or 3/$79.95**
NEC8KR (NEC PC-8201A). . . . . **$29.95 ea. or 3/$79.95**
OM108K (Olivetti M10). . . . . **$29.95 ea. or 3/$79.95**

TANDY 200

Easy to install module plugs right into the socket increasing memory in 24K increments. Complete with module and documentation for installation.

M200R (Tandy 200 Expansion). . . . . **$99.95 ea. or 2/$189.95**

TRS-80 MODEL I AND III

Each Kit comes complete with 8-MM5290 (UPD416/4116) 16K Dynamic RAMs and documentation for conversion. Model I: 16K equipped with Expansion Interface can be expanded to 48K with 2 Kits. Model III: Can be expanded from 16K to 48K using 2 Kits. Each Kit will expand computer by 16K increments.

TRS-16K3 200ns (Model III). **$5.95**
TRS-16K4 250ns (Model I). **$5.49**

UPDATE! TRS-80 MODEL IV, 4P UPDATE!

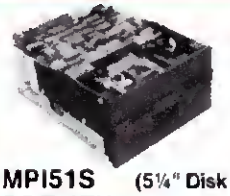
Easy to install Kits come complete with: TRS-64K-2 (8 each 4164N-20 (200ns) 64K Dynamic RAMs); TRS-64K-2PAL (8 each 4164's plus PAL chip) and documentation for conversion.

TRS-64K-2 Expands Model IV from 16K-64K or Model IV & 4P from 64K-128K (w/Mem. Disk). **$9.95**
TRS-64K-2PAL Expands Model IV (w/Black & White Monitors only) from 64K-128K. **$29.95**

TRS-80 COLOR AND COLOR II

Easy to install Kit come complete with 8 each 4164-20 (200ns) 64K Dynamic RAMs and documentation for conversion. Converts TRS-80 Color Computers with D, E, ET, F and NC circuit boards to 32K. Also converts TRS-80 Color Computer II to 64K. Flex DOS or OS-9 required to utilize full 64K RAM on all computers.

TRS-64K-2. . . . . . **$9.95**



MPI 5¼" Disk Drive
TRS-80 Model I and III

- Use as a second disk drive • Single-sided
- Single/double density • Full height drive
- 48 TPI • Documentation incl. • Wt.: 3.7 lbs.

MPI51S (5¼" Disk Drive). . . . . **$69.95 each**
DDE-1FH (5¼" Disk Drive Enclosure). . . . . **$69.95 each**

We also specialize in integrated circuits, custom cables, power supplies, keyboards, and much, much more! Give us a call today!

$20.00 Minimum Order-U.S. Funds Only. CA Residents Add 6% or 6½% Sales Tax. Spec Sheets-30¢ ea. Shipping—Add 5% + $1.50 Insurance. Prices Subject to Change. *Send S.A.S.E. for Quarterly Sales Flyer!* Send $1.00 Postage for your *FREE 1986 JAMECO CATALOG!*

MasterCard — VISA

Mail Order Electronics - Worldwide
Jameco ELECTRONICS

1355 SHOREWAY ROAD, BELMONT, CA 94002
12/85 *Phone Orders Welcome (415) 592-8097 Telex: 176043*

Give A Gift That Will Be Opened Every Month

This Season, Give 80 Micro

Why give a present that will be opened just once when you can give one that will be opened year-round? You'll be remembered this season, and every season, when you send the perfect gift for TRS-80* users—an **80 Micro** subscription.

Month after month, **80 Micro** will be a valuable, practical, and sometimes surprising guidebook for the dedicated TRS-80 user. And for an incredibly low price, it offers more assistance and inspires more efficiency than most peripherals and software packages. Just look at the benefits:

- Shop-at-home convenience with concise new product information.
- Answers to both common and unusual computing problems in "Feedback Loop".
- Thorough evaluations and quality ratings of the latest Tandy-compatible products.

Plus, a variety of type-and-run program listings, up-to-the-minute industry trends, and much more.
All in the **only** system-specific source on the market for TRS-80 users.

All for just $24.97—12 issues at **48% off** the cover price. And you won't be billed until **after** the holidays.

Send an All Seasons Greetings to someone special this season by returning the coupon, or by calling 1-800-258-5473. (In NH, dial 1-924-9471.)

YES! **I Want To Give A Year-Round Gift.**

Please send an **80 Micro** gift subscription to the person listed below. I'll pay $24.97 for 12 issues—a 48% savings.

☐ Payment Enclosed ☐ Bill me after the holidays!

Please make checks payable to **80 Micro**.

Name of Gift Recipient _____

Address _____

City _____ State _____ Zip _____

My Name _____

Address _____

City _____ State _____ Zip _____

Canada & Mexico $27.97, 1 year only, US funds drawn on US bank.
Foreign Surface $44.97, 1 year only, US funds drawn on US bank. Foreign Airmail, please inquire.
All gift subscriptions will begin with the first available issue in 1986.

* TRS-80 is a registered trademark of Radio Shack, a division of Tandy Corporation.

65DB8

Listing 2 continued

```
F03D DD7700    00043          LD      (IX),A          ;Save it in buffer
F040 ED78      00044          IN      A,(C)           ;get byte 2
F042 DD7701    00045          LD      (IX+1),A        ;save it
F045 ED78      00046          IN      A,(C)           ;get byte 3-8
F047 DD7702    00047          LD      (IX+2),A        ;& store them
F04A ED78      00048          IN      A,(C)
F04C DD7703    00049          LD      (IX+3),A
F04F ED78      00050          IN      A,(C)
F051 DD7704    00051          LD      (IX+4),A
F054 ED78      00052          IN      A,(C)
F056 DD7705    00053          LD      (IX+5),A
F059 ED78      00054          IN      A,(C)
F05B DD7706    00055          LD      (IX+6),A
F05E ED78      00056          IN      A,(C)           ;Read 8th byte
F060 DD7707    00057          LD      (IX+7),A        ;Save it
F063 0608      00058          LD      B,8             ;B= 4 of bytes read
F065 1680      00059 ROTATE   LD      D,128           ;D is mask bit
F067 AF        00060          XOR     A               ;Zero A
F068 DDCB0006  00061          RLC     (IX)            ;Rotate byte 1
F06C 3001      00062          JR      NC,$+3          ;Bit set?
F06E B2        00063          OR      D               ;If so set same bit on A
F06F CB0A      00064          RRC     D               ;D is now bit 6
F071 DDCB0106  00065          RLC     (IX+1)          ;Rotate Byte 2
F075 3001      00066          JR      NC,$+3          ;Is it set
F077 B2        00067          OR      D               ;If so Set bit on A
F078 CB0A      00068          RRC     D               ;Set bit 5 on D
F07A DDCB0206  00069          RLC     (IX+2)
F07E 3001      00070          JR      NC,$+3
F080 B2        00071          OR      D
F081 CB0A      00072          RRC     D
F083 DDCB0306  00073          RLC     (IX+3)
F087 3001      00074          JR      NC,$+3
F089 B2        00075          OR      D
F08A CB0A      00076          RRC     D
F08C DDCB0406  00077          RLC     (IX+4)
F090 3001      00078          JR      NC,$+3
F092 B2        00079          OR      D
F093 CB0A      00080          RRC     D
F095 DDCB0506  00081          RLC     (IX+5)
F099 3001      00082          JR      NC,$+3
F09B B2        00083          OR      D
F09C CB0A      00084          RRC     D
F09E DDCB0606  00085          RLC     (IX+6)
F0A2 3001      00086          JR      NC,$+3
F0A4 B2        00087          OR      D
F0A5 CB0A      00088          RRC     D
F0A7 DDCB0706  00089          RLC     (IX+7)
F0AB 3001      00090          JR      NC,$+3
F0AD B2        00091          OR      D
F0AE 77        00092          LD      (HL),A  ;A= New Byte, Store it
F0AF CDD9F0    00093          CALL    PRINTA  ;Print Byte
F0B2 10B1      00094          DJNZ    ROTATE  ;Do 8 in all
F0B4 3A10F1    00095          LD      A,(POSX)        ;Get x position
F0B7 3C        00096          INC     A               ;Inc it
F0B8 3210F1    00097          LD      (POSX),A        ;Save it
F0BB C1        00098          POP     BC              ;Get column count
F0BC 05        00099          DEC     B               ;Decrement
F0BD AF        00100          XOR     A               ;Zero A
F0BE B8        00101          CP      B               ;Done Last Column?
F0BF C230F0    00102          JP      NZ,MIDDLE       ;Go if not
F0C2 360A      00103          LD      (HL),0AH        ;Print a Line feed
F0C4 CDD9F0    00104          CALL    PRINTA
F0C7 3A11F1    00105          LD      A,(POSY)        ;Get Y position
F0CA C608      00106          ADD     A,8             ;Add 8
F0CC 3211F1    00107          LD      (POSY),A        ;save it
F0CF C1        00108          POP     BC              ;Get count
F0D0 05        00109          DEC     B               ;Decrement count
F0D1 AF        00110          XOR     A               ;A=0
F0D2 B8        00111          CP      0               ;Check count
F0D3 C223F0    00112          JP      NZ,OUT          ;Cont. not zero
F0D6 C3F1F0    00113          JP      BYE             ;Quit if Zero
               00114 ;-----Print Routine
F0D9 DBFB      00115 PRINTA   IN      A,(251)         ;Get printer Status
F0DB FE3D      00116          CP      61              ;Is it ready?
F0DD 20FA      00117          JR      NZ,PRINTA       ;If not wait
F0DF 7E        00118          LD      A,(HL)          ;Get char in HL
F0E0 D3FB      00119          OUT     (251),A         ;Send to Printer
F0E2 C9        00120          RET                     ;Return
               00121 ;-------Put Printer in Graphics Mode
F0E3 E5        00122 GRAMOD   PUSH    HL              ;Save HL
F0E4 2107F1    00123          LD      HL,BGMODE       ;Get Printer Codes
F0E7 0604      00124          LD      B,4             ;4 bytes
F0E9 CDD9F0    00125 GRA001   CALL    PRINTA          ;Print them
F0EC 23        00126          INC     HL              ;Next byte
F0ED 10FA      00127          DJNZ    GRA001          ;Repeat
F0EF E1        00128          POP     HL              ;Get HL
F0F0 C9        00129          RET                     ;Return
               00130 ;-------End Program/Return Printer to Normal
F0F1 210BF1    00131 BYE      LD      HL,EGMODE       ;Get Printer Codes
F0F4 0602      00132          LD      B,2             ;2 bytes
F0F6 CDD9F0    00133 BYE2     CALL    PRINTA          ;Print One
F0F9 23        00134          INC     HL              ;Get next
F0FA 10FA      00135          DJNZ    BYE2            ;Go print it
F0FC 3EFC      00136          LD      A,0FCH  ;No inc/dec, waits, board off
F0FE D383      00137          OUT     (STATUS),A      ;Set Graphics Options
F100 DDE1      00138          POP     IX              ;Get regs
F102 C1        00139          POP     BC
F103 D1        00140          POP     DE
F104 E1        00141          POP     HL
F105 AF        00142          XOR     A               ;Zero A
F106 C9        00143          RET                     ;Return to Caller
```

Listing 2 continued

Listing 2 continued

```
                00144 ;-------Data
8003            00145 STATUS   EQU     83H
F107 1B         00146 BGMODE   DEFB    27
F108 4C         00147          DEFB    'L'
F109 80         00148          DEFB    128
F10A 02         00149          DEFB    2
F10B 1B         00150 EGMODE   DEFB    27
F10C 40         00151          DEFB    '@'
F10D 1B         00152 NUMPIN   DEFB    27
F10E 41         00153          DEFB    'A'
F10F 08         00154          DEFB    8
F110 00         00155 POSX     DEFB    0
F111 00         00156 POSY     DEFB    0
0008            00157 SCRBUF   DEFS    8
0002            00158 PRTBUF   DEFS    2
                00159 ;-------Initialize Graphics Board-Routine found in Manual
                00160 ; Insert lines 125-154 of the Initialization routine
                00161 ; page 89-90 of the TRS-80 Computer Graphics Operation
                00162 ; Manual.  Radio Shack Catalog # 26-1125
F000            00190          END     GPRINT
```

End

```
1770 IF FF<>28 THEN 1790 ELSE POKE 128,135:GOSUB 1970 :PUT(X1,Y1
),CU,XOR :VIEW(0,0)-(639,239) :CLS :SCREEN1 :INPUT"READY PRINTER
 & PRESS 'ENTER'";ZZ$ :GOSUB 1990 :IF PEEK(128)=135 THEN SYSTEN
PR$ ELSE FI$=PR$ + CHR$(13) :GOSUB 2
1790 IF FF<>29 TBEN 1810 ELSE POKE 128,135:GOSUB 1970 :VIEW(0,0)
-(639,239) :CLS :SCREEN1 :INPUT"ENTER FILENAME";FI$ :GOSUB 1990
:FI$=GSAVE "+FI$+ CHR$(13) :GOSUB 2
1810 IF FF<>30 THEN 1B3B
182B PUT(X1,Y1),CU,XOR :VIEW(0,0)-(639,239) :CLS :GOSUB 197B :SC
REEN1 :PRINT :PRINT :INPUT"ENTER FILENAME";FI$ :FI$="GLOAD " +FI
$+CHR$(13) :GOSUB 2 :RC=1: GOSUB 2340 :RC=0 :SCREEN0 :POKE128,13
4 :GOTO188
```

Figure. Change Rembrandt's lines to those listed above.

The answers to the TRS-80 trivia quiz (Sidetracks, p. 8):

1. The Model I with Level I Basic: WHAT?, HOW?, and SORRY.
2. Start up an Exatron Stringy-Floppy.
3. RVEJARAJ.
4. The sentence reads, "Joe, you rummy buzzard!" It was used in a prototype format program's verification utility and got inserted at the end of each sector in the early TRSDOS 1.3 disks. The format program was later changed to insert "(c) 1980 Radio Shack."
5. The Inventory Control program for Radio Shack stores, modified to feed itself nonsense data.
6. International Jewelers Guild.
7. TRSDOS 1.3.
8. ULTRADOS.
9. A$ and B$; 16 characters.
10. It was a tape-based program that used Disk Basic verbs for mostly graphics-oriented functions. Its features included a timed input statement, where you could set a time interval in which response had to occur before the program took a branch elsewhere.

TIDBIT #29

It isn't easy to hide information like a serial number or surprise message in Basic program lines. Here's a quick and dirty way to give Model I/III Basic code a little privacy; it depends on a quirk in the Basic line editor. Follow these seven steps carefully:

1. Type in your line of Basic code as usual. Shorter lines work best; you must have some room at the end of the line. Press the enter key.
2. Get into Basic's edit mode by typing in EDIT and the line number, then press the enter key.
3. Press the "X" key to get to the end of the line.
4. Extend the line by typing in a colon and REM (:REM) or a colon and an apostrophe (:').
5. Press and hold the shift key. At the same time, press the left-arrow key. Each time you press the left arrow, the cursor backspaces once without erasing the character under it. Backspace to the beginning of the message or code you want to hide.
6. Now type in a new message. This will cover the original code. For example, you might want to cover GOTO 500 with PRINT A$. You can use spaces if you can't think of anything else.
7. Press the enter key to lock in the new code. When you list the line, the display shows only the new information. Actually, Basic displays the original code and immediately covers it up; you should avoid long cover-ups as they might jitter on the screen.

When would you use this technique? You might want to hide a serial number contained in a program. If the original code were SN = 25, you could cover the 25 with 38. A user of the program would see the second number but Basic would use the first. The Remark statement prevents any of the cover-up code from executing.

You could cover a GOTO 500 with GOSUB 650—have fun following that program! Or you could hide a surprise message by covering :PRINT "YOU LOSE, TURKEY" with :REM END OF GAME ROUTINE. Or try hiding something like IF N$ = "ANDY" THEN CMD"S", where N$ is the name of a friend. Your friend will be puzzled because the program will list properly but will always seem to reboot—on him or her only.

Andy Levinson
Studio City, CA

Trying to read a long file as it whizzes by your screen is a study in frustration. To properly inspect file lines, you need a program that writes files to your screen in small, controllable pieces. Page, a Model 4 Assembly-language program, displays files either a screenful at a time or line by line.

Type in the code in Basic, run it, and it will write the file Page/CMD to disk. To use Page, type in PAGE FILE NAME at TRSDOS Ready. Page lists a screenful of the indicated file's code and pauses for a response. Pressing the spacebar writes the next screenful of code, and pressing the enter key writes the file's next line.

If you press the break key or control-C, you exit the program. Of course, Page also stops at the end of the file. Since I designed this program for standard text files only, you must save Basic programs in ASCII format, using the SAVE FILE NAME, A option.

Dan Velting
Kentwood, MI

Editor's note: We have published the accompanying listing in Basic data statements. The Basic program and /CMD file are available on Load 80.

Program Listing. Page.

```
10 OPEN "O",1,"PAGE/CMD"
20 FOR I% = 1 TO 652
30 READ X%
40 PRINT #1, CHR$(X%);
50 NEXT I%
60 CLOSE 1
70 END
100 DATA 1,2,0,48,0,205,119,84,212,141,84,220,157,84,205,119,84,195,141
110 DATA 84,205,157,84,14,6,205,119,84,212,141,84,220,157,84,201,33,168,84
120 DATA 9,78,35,70,35,94,66,106,241,201,205,65,73,205,115,58,205,232,57
130 DATA 254,34,40,4,254,39,32,110,71,205,219,57,205,219,57,42,160,185,43
140 DATA 43,229,184,14,0,40,11,12,205,219,57,254,13,40,9,184,32,245,205
150 DATA 164,57,184,40,239,205,232,57,205,115,58,254,44,40,9,254,59,40,5
160 DATA 184,40,2,254,13,225,34,160,185,32,58,121,254,2,56,45,205,214,57
170 DATA 71,205,219,57,205,219,57,24,23,85,115,97,103,101,58,32,112,97,103
180 DATA 101,32,102,105,160,101,13,0,40,8,205,164,57,254,44,40,135,201,205
190 DATA 161,68,24,129,205,229,61,245,58,60,106,183,258,1,71,241,205,37,69
200 DATA 121,254,44,262,4,65,201,205,67,78,245,120,230,131,196,154,51,241,201
210 DATA 205,65,73,205,115,50,254,34,40,5,254,39,196,114,51,74,175,245,205
220 DATA 219,57,254,13,202,1,71,185,32,6,205,164,57,185,32,10,71,241,197
230 DATA 183,196,161,60,241,24,220,193,245,120,135,120,40,1,2,6,49,5,246
240 DATA 128,205,161,68,241,254,44,40,197,201,205,65,73,205,34,86,205,225,84
250 DATA 121,254,44,32,9,213,205,229,61,67,209,195,3,86,58,231,108,103,40
260 DATA 22,6,0,122,179,200,27,42,44,106,35,34,44,106,58,70,105,103,196
270 DATA 45,74,24,236,42,44,106,25,58,43,106,71,195,52,84,205,153,85,58
280 DATA 70,105,183,192,58,107,106,254,85,192,60,50,107,106,201,205,65,73,205
290 DATA 225,84,205,67,70,205,24,70,121,254,44,40,239,201,58,77,105,61,250
300 DATA 110,86,71,50,77,105,58,78,105,61,184,32,3,50,78,105,58,78,105
310 DATA 144,159,60,50,130,105,205,153,86,50,79,105,195,242,57,205,119,51,201
320 DATA 58,79,17,81,40,6,8,126,254,13,202,179,49,254,32,202,179,49,10
330 DATA 35,19,4,195,161,49,62,13,18,120,254,0,194,194,49,62,1,183,195
340 DATA 197,49,62,0,183,194,80,58,33,81,48,17,105,40,62,78,239,33,156
350 DATA 40,6,4,62,59,239,202,226,49,79,203,241,62,26,239,33,0,40,17
360 DATA 105,48,6,60,62,3,239,194,0,50,230,127,119,35,254,13,202,251,49
370 DATA 16,239,54,0,62,0,183,1,126,0,50,194,101,50,33,0,40,70,121
380 DATA 6,8,184,202,21,50,62,2,239,35,202,6,50,58,137,40,61,50,137
390 DATA 40,58,137,48,6,0,184,204,40,50,195,226,49,62,1,239,254,32,202
400 DATA 66,50,254,13,282,72,50,254,128,202,106,50,254,3,202,106,50,195,40
410 DATA 50,62,23,50,137,48,201,58,137,48,60,50,137,48,201,33,138,48,78
420 DATA 121,6,0,184,202,90,50,62,2,239,35,202,83,50,195,121,50,254,28
430 DATA 194,115,50,17,105,48,62,60,239,202,121,50,79,203,241,62,26,239,62
440 DATA 22,239,2,2,156,49
```

Making TRS-80 Assemblers Toe the Hex/ASCII Line

When you build a hardware project, your software has to work with it. But the output from TRS-80 assemblers, such as EDTASM and ALDS, doesn't conform to the industry-standard Intel hexadecimal/ASCII absolute object code format, which lets you easily load and transfer microprocessor object files.

Since I use the hex/ASCII format for much of my development debugging equipment, I wrote a program that converts TRS-80 object files to hex/ASCII. I can download such files to my emulator (such as the Huntsville Microsystems Z80 emulator in the Photo) and Sunrise Electronics EPROM programmer. The DR800 single-board computer in the April (p. 82) and May (p. 78) columns also accepts code in the hex/ASCII format.

I can also easily send them over telephone lines using a modem. Hex/ASCII has several advantages. First, it includes object location (addressing) information so the system must know where in memory to put the code. Second, it includes a data integrity check (checksum) so you can transfer data reliably to another system. Finally, it uses only printable ASCII characters and a carriage return at the end of each line, avoiding special control characters that the receiving system might not understand.

Photo. The Huntsville Microsystems Z80 emulator.

TRS-80 Absolute Object File Format

Before describing the Intel hex/ASCII format, I'll discuss the TRS-80 absolute object file format. This is essential to understanding my conversion program.

I have experience with the formats produced by Radio Shack's EDTASM editor/assembler (running under NEWDOS/80) and with Radio Shack's Assembly Language Development System (ALDS), which I use on my Model 4P. In general, EDTASM and ALDS generate the same format for an executable object file, but ALDS' format is slightly different under certain conditions.

Figure 1 shows the general format for Radio Shack's absolute object files. The first byte is a record header and is always a 1 (01 hex) as long as the file has at least 1 byte of code. The second byte is a count value, indicating the number of data bytes in the record plus the number of address bytes (there are always 2 address bytes). The next 2 bytes indicate the starting memory address for the object bytes in the current record; the low-order byte is first, followed by the high-order byte. The address bytes are then followed by the specified number of data (object code) bytes, which are to be placed into memory. All values are in binary, not ASCII.

As many of these 01 hex type records follow as is necessary to hold all of the object code bytes. Once all of the bytes have been included in these records, the assembler puts a terminator at the end

	Header byte	Record byte count	Record starting address (low)	Record starting address (high)	Record data (nn – 2 bytes)
Record 1	01	nn	sl	sh	xx . . . xx
	01	nn	sl	sh	xx . . . xx
			.		
			.		
			.		
Record n	01	nn	sl	sh	xx . . . xx
Terminator Record	02	02	sl	sh	

Program starting execution address

Figure 1. TRS-80 absolute object file format. (All values are in hex.)

LOAD 80

System Requirements

Models I and III with changes
Model 4
Disk Basic

Photograph by Frank Cordelia

of the file. The first byte of the terminator record, the record header, is always a 2 (02 hex), as is the second byte (the record byte count). The final 2 bytes of the record, bytes 3 and 4, are the execution starting address of the program, which can be (and often is) different from the starting address where the object code is loaded into memory. Again, the address bytes are in low-byte/high-byte order.

ALDS modifies this format slightly if you assemble your program absolutely (by specifying the starting address as an operand to the PSECT pseudo-op and avoiding program-linking): it adds one record at the beginning of the file. The record's format is shown in Fig. 2. This is not clearly indicated in the ALDS manual. The record header is a 5 (05 hex) byte. This is followed by a record byte count byte, which is followed by the specified number of data bytes. The program's starting address is, however, included in this record (bytes 4 and 5), though it seems redundant, since it is also in the initial data record. You can discard this extra record without losing any information.

A sample Z80 Assembly-language program in Program Listing 1 (from ALDS) is written like an interrupt service routine. It saves the CPU registers by swapping register banks, causes a time delay by counting down a value in the HL register pair, and then restores the registers and enables interrupts before exiting via a Return instruction.

This sample program is 14 bytes long, and specifies the label DELAY as the execution starting location. If you assemble and link the file starting at address 7000 hex, the label DELAY is assigned the value 7002 hex and the absolute object file generated by ALDS looks like that shown in Fig. 3. Note that it is a binary file and all values are given in hex.

Header byte	Record byte count	Record information (nn bytes)
05	nn	xx . . . xx

Figure 2. ALDS assembler object file record addition for absolutely assembled programs. (All values are in hex—base 16.)

The Intel Hex/ASCII Format

While the TRS-80 format is adequate for many uses, it isn't flexible enough for general-purpose object files. In particular, it lacks a checksum and a way to easily transfer files. While an internal system checksum verifies the object information as it loads from the disk, there isn't one for transferring files.

When you transfer the object file from one computer to another over an RS-232C serial line, the receiving system probably tries to interpret some of the bytes being received, since many of them are defined as ASCII control characters.

To avoid this problem, the hex/ASCII format contains only printable ASCII characters (except for the carriage return at the end of lines, as mentioned earlier).

As in the TRS-80 object file format, the hex/ASCII format has two basic record types: data records and a terminator record. Figure 4 shows the formats for the hex/ASCII data and terminator records. Note that all characters and object information are now ASCII characters, not binary values shown in hex. Each byte of information in the data record is actually stored as 2 hex/ASCII bytes in memory, in the hex range of zero to 9 and A to F. For example, the bytes 38 90 BA 2C would be stored in memory (or on disk)

```
 Tandy Corp. ALOS ALASM copr. 1982,83 v.03.02.00  Page 1          09/27/85
Assembly Listing of NEXASCII/SRC:1


E Addr     Obj          Fl  Ln #          Source Line

  0000'                    00001  EXAMPL  PSECT
                           00004  ;       LAST MODIFICATION DATE: 09/25/85
                           00005  ;
                           00006  ;*****************************************************************
                           00007  ; FILE: HEXASCII/SRC
                           00008  ;
                           00009  ; AUTHOR: Roger C. Alford
                           00010  ;
                           00011  ; MODULE DESCRIPTION:
                           00012  ;       This program is merely an example program for the Project 80
                           00013  ;       discussion of Intel Hex/ASCII object code format.
                           00014  ;
                           00015  ;
  0000' D9               -  00016  EXAMPL  EXX                      ;SWAP THE MAIN CPU REGISTERS
  0001' 08                   00017            EX      AF,AF'         ;SWAP AF TO SAVE IT TEMPORARILY
  0002' 213412               00018  DELAY     LD      HL,1234H       ;LOAD THE DELAY COUNT VALUE INTO HL
  0005' 2B                   00019  LOOP      DEC     HL             ;DECREMENT THE DELAY LOOP COUNT VALUE
  0006' 7C                   00020            LD      A,H            ;IS THE COUNT VALUE ZERO YET?
  0007' B5                   00021            OR      L
  0008' 20FB                 00022            JR      NZ,LOOP        ;IF NOT, LOOP AGAIN
  000A' 08                   00023            EX      AF,AF'         ;RESTORE REGISTERS A AND F
  000B' D9                   00024            EXX                    ;REGISTER THE MAIN CPU REGISTERS
  000C' FB                   00025            EI                     ;ENABLE Z80 MASKABLE INTERRUPTS
  000D' C9                   00026            RET                    ;RETURN FROM THIS SERVICE ROUTINE
                           00027  ; END OF SERVICE ROUTINE: EXAMPL.
                           00028  ;
  0002'                    00029            END     DELAY          ;START EXECUTION AT LOCATION 'DELAY'


    No Assembly Errors

Time  = 0:01
Bytes = 14
Lines = 27
```

Program Listing 1. Sample Z80 Assembly-language program.

as 33H 38H 39H 30H 42H 41H 32H 43H.

Notice how 2 bytes are stored in memory for each information byte in the hex/ASCII data record. This is the hex/ASCII format's main disadvantage: it isn't very memory efficient.

The first character in every hex/ASCII record is the colon (:); it has a value of 3A hex and is the start-of-record indicator character. The first byte (two ASCII characters) following the colon indicates the number of data bytes in the record (all values are in hex). The next 2 bytes indicate the starting memory address for the data bytes in that record (high byte first). The fourth byte is the record type indicator, which is always 00 for a data record and 01 for a terminator record.

The object data for the record, if any, follows the record type indicator byte.

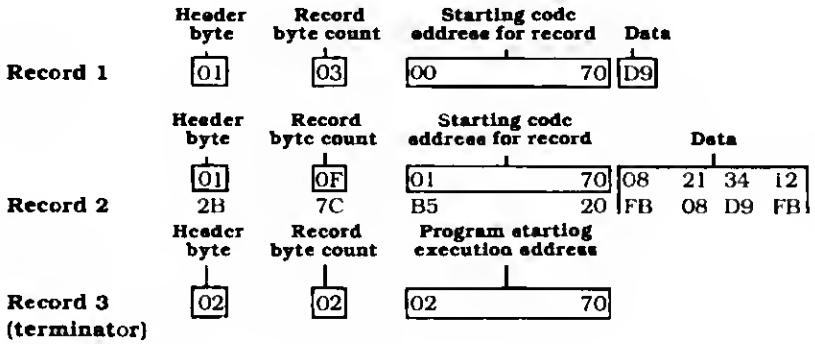


Figure 3. ALDS object file output for Listing 1 program. (All values are in hex—base 16.)

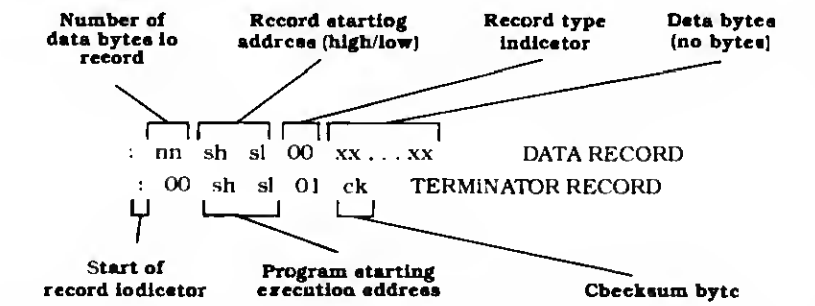


Figure 4. Intel hex/ASCII record formats. (All characters are ASCII. Spaces between record bytes are shown only for clarity and do not exist in the actual records.)

```
:01700000D9B6
:0D7001000821341228B7CB520FB08D9FBC9F7
:007002018D
```

Figure 5. Hex/ASCII object file printout for program in Program Listing 1.

TRS-80 filespec entered	Conversion program interpretation
TESTFILE	TESTFILE/CMD:0
TESTFILE/	TESTFILE:0
PROJCT80/:1	PROJCT80:1
PROJCT80:2	PROJCT80/CMD:2
MICRO80/ABS	MICRO80/ABS:0
MICRO80/ABS:1	MICRO80/ABS:1

Figure 6. Interpretation of entered object file filespecs for hex/ASCII conversion program.

Terminator records have no data bytes, but data records should always have at least 1 data byte. A checksum byte follows the data bytes. When all of the bytes in the record are added together, including the checksum byte (ignoring any carries above 8 bits), the result is 00. The checksum totals include only the hex values displayed, not the ASCII numeric values. For example, a data record containing :0100040023 (all characters are ASCII) would have the checksum D8, since 01 + 00 + 04 + 00 + 23 + D8 = 00 (carry ignored).

The resulting final data record is :0100040023D8 (again, with all characters in ASCII).

The terminator record always has a 00 length specification, as mentioned above, since the terminator record includes no data bytes. The address value included in the terminator record specifies the execution starting address for the program.

With the conversion program, you can generate the hex/ASCII object file for the example program in Listing 1 with a printout (Fig. 5).

One final note about the hex/ASCII format. Systems reading in or receiving object information from a hex/ASCII file are supposed to look for the start-of-record character (colon). All characters before the first colon are to be ignored. Thus you can store information in the hex/ASCII object file before the data records. You can use this feature to store the symbol table for the program (with only ASCII characters and no colons, of course), which lets you load the symbol table with the object file for symbolic debugging.

The Conversion Program

The conversion program (Program Listing 2) is in Model 4 Basic. It will run under Model I/III Basics if you change the long variable names to one- or two-character names and change the INSTR functions in lines 40, 100, and 160 to subroutine calls. You can do this by using the assignment SV$ = ":" or SV$ = "/" (whichever is appropriate) and calling the subroutine in Program Listing 3. For example, line 40 would become

```
40 SV$ = ":":GOSUB 8000:DRVPOS = SV.
```

You might also need to change line 10030, which returns you to DOS, depending on what DOS you're using.

Lines 5–12 are the comment header and startup message for the program. Lines 13–14 dimension and initialize the HEXVAL$ array, which holds the 16 hex/ASCII characters in numerical order.

Lines 20–340 acquire the filespec for the TRS-80 object file and determine the filespec of the output hex/ASCII file. You can enter the TRS-80 object filespec in

one of several ways. If you don't include an extension, the default is /CMD. If you include the file name with a "/" suffix, without any extension characters, the program will assume that the file name has no extension. Or you can give the file an extension of your choice. The drive is zero unless you specify otherwise. Figure 6 shows several possible filespecs, along with the actual filespec interpretation by the conversion program.

The hex/ASCII output file has the same filespec as the input file, except that it gets a /HEX extension. You're prompted for the drive number for the hex/ASCII file; press the enter key for the default drive (the same number as the input file) which is in parentheses, or enter the desired drive number.

Lines 400–550 initialize the variables and open the files. The variable TOTALBYTECOUNT is the accumulator to count the total number of data (object code) bytes in the file. OBJFILE$ is the input file and HEXFILE$ is the output file. The input file is a random-access file with a record length of one, whereas the output file is a sequential file. OBJDATA$ stores the input records.

Lines 560–1340 do most of the file processing. Lines 592–598 cause the extra ALDS record (with the 05 hex header byte) to be ignored, if present. The program converts the remaining object data to hex/ASCII format and writes it to the output file. Lines 2000–2280 generate the terminator record, close the files, and exit through line 10030, which is currently a return to DOS.

The program displays the number of object code bytes, along with the program's starting execution address.

Lines 3000–3040 add the decimal value in DECNUM to the running checksum accumulator, CHKSUM.

The subroutine at lines 4000–4100 converts the decimal (base 10) value in DECNUM to a hex/ASCII character in HEXNUM$, using the HEXVAL$ array.

The subroutine at lines 5000–5060 increments the address variables ADDRL and ADDRH, which keep track of the current object byte address for the hex/ASCII file. The subroutine at lines 6000–6100 calculates the checksum byte for the current hex/ASCII record and writes it to the output file.

The subroutine at lines 9000–9020 retrieves the next byte from the input file, and updates the input file record pointer, OBJPTR%. Lines 10000–10020 generate a data read error message and close the files if a data error is detected.■

Write to Roger C. Alford at P.O. Box 2014, Ann Arbor, MI 48106. Please enclose a self-addressed, stamped envelope for a reply.

Program Listing 2. Model 4 hex/ASCII conversion program.

```
5 '*****************************************************************
6 ' THIS PROGRAM CONVERTS ALDS ABSOLUTE OBJECT FILES TO INTEL HEX/ASCII *
7 ' FORMAT. THE OUTPUT FILENAME IS THE SAME AS THE INPUT FILENAME, BUT  *
8 ' HAS THE EXTENSION "/HEX".                                           *
9 '          +++ CREATED BY Roger C. Alford   08/04/85 +++              *
10 '*****************************************************************
11 CLS:PRINT "TRS-80 BINARY TO INTEL HEX/ASCII FORMAT CONVERSION PROGRAM"
12 PRINT "       by Roger C. Alford       Version 1.2  08/25/85":PRINT
13 DIM HEXVAL$(16):FOR I%=0 TO 15:READ HEXVAL$(I%):NEXT I%
14 DATA "0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F"

20 INPUT "ENTER OBJECT FILENAME (/CMD) ";OBJFILE$
30 IF LEN(OBJFILE$)=0 THEN 20
40 DRVPOS=INSTR(OBJFILE$,":")
60 IF DRVPOS=0 THEN OBJDRIVE$=":0":GOTO 100 ELSE OBJDRIVE$=MID$(OBJFILE$,DRVPOS,
2)
80 OBJFILE$=LEFT$(OBJFILE$,DRVPOS-1)
100 EXTPOS=INSTR(OBJFILE$,"/")
120 IF EXTPOS=0 THEN OBJFILE$=OBJFILE$+"/CMD":GOTO 160
140 IF EXTPOS=LEN(OBJFILE$) THEN OBJFILE$=LEFT$(OBJFILE$,EXTPOS-1)
160 EXTPOS=INSTR(OBJFILE$,"/")
180 IF EXTPOS<>0 THEN HEXFILE$=LEFT$(OBJFILE$,EXTPOS-1) ELSE HEXFILE$=OBJFILE$
200 HEXFILE$=HEXFILE$+"/HEX"
220 OBJFILE$=OBJFILE$+OBJDRIVE$
240 PROMPT$="ENTER DRIVE NUMBER TO STORE HEX FILE ("+RIGHT$(OBJDRIVE$,1)+") "
260 PRINT PROMPT$;
280 INPUT HEXDRIVE$
300 IF LEN(HEXDRIVE$)=0 THEN HEXDRIVE$=OBJDRIVE$:GOTO 340
320 IF LEN(HEXDRIVE$)=1 THEN HEXDRIVE$=":"+HEXDRIVE$ ELSE GOTO 260
340 HEXFILE$=HEXFILE$+HEXDRIVE$

400 TOTALBYTECOUNT=0
500 OPEN "R",1,OBJFILE$,1
520 FIELD 1,1 AS OBJDATA$
540 OBJPTR%=1
550 OPEN "O",2,HEXFILE$

560 GOSUB 9000
570 OBJRECTYPE=ASC(OBJDATA$)
580 IF OBJRECTYPE<>1 AND OBJRECTYPE<>2 AND OBJRECTYPE<>5 THEN 10000
585 GOSUB 9000
590 OBJRECLEN=ASC(OBJDATA$)-2
592 IF OBJRECTYPE<>5 THEN 600
594 FOR I%=1 TO OBJRECLEN+2
596 GOSUB 9000:NEXT I%
598 GOTO 560
600 GOSUB 9000
620 ADDRL=ASC(OBJDATA$)
640 GOSUB 9000
660 ADDRH=ASC(OBJDATA$)
680 IF OBJRECTYPE=2 THEN 2000
1000 TOTALBYTECOUNT=TOTALBYTECOUNT+OBJRECLEN
1010 IF OBJRECLEN>=16 THEN DATACNT=16 ELSE DATACNT=OBJRECLEN
1020 PRINT #2,":";
1040 CHKSUM=0
1060 DECNUM=DATACNT:GOSUB 3000:GOSUB 4000
1080 PRINT #2,HEXNUM$;
1100 DECNUM=ADDRH:GOSUB 3000:GOSUB 4000
1120 PRINT #2,HEXNUM$;
1140 DECNUM=ADDRL:GOSUB 3000:GOSUB 4000
1160 PRINT #2,HEXNUM$;
1180 PRINT #2,"00";
1200 FOR I=1 TO DATACNT
1220 GOSUB 9000:OBJRECLEN=OBJRECLEN-1
1240 DECNUM=ASC(OBJDATA$):GOSUB 3000:GOSUB 4000
1260 PRINT #2,HEXNUM$;
1280 GOSUB 5000
1300 NEXT I
1320 GOSUB 6000
1340 IF OBJRECLEN<>0 THEN 1010 ELSE 560

2000 PRINT #2,":00";
2020 CHKSUM=0
2040 DECNUM=ADDRH:GOSUB 3000:GOSUB 4000
2060 ADDRH$=HEXNUM$
2080 PRINT #2,HEXNUM$;
2100 DECNUM=ADDRL:GOSUB 3000:GOSUB 4000
2120 ADDRL$=HEXNUM$
2140 PRINT #2,HEXNUM$;
2160 PRINT #2,"01";
2170 DECNUM=1:GOSUB 3000
2180 GOSUB 6000
2200 CLOSE
2220 PRINT:PRINT "THE TOTAL NUMBER OF PROGRAM BYTES IS: ";TOTALBYTECOUNT
2240 PRINT "THE EXECUTION STARTING ADDRESS IS: ";
2260 PRINT ADDRH$;ADDRL$;" (HEX)":PRINT
2280 GOTO 10030

3000 '*****************************************************************
3010 ' THIS SUBROUTINE ADDS THE "DECNUM" VALUE TO "CHKSUM"
3020 CHKSUM=CHKSUM+DECNUM
3040 RETURN

4000 '*****************************************************************
4010 ' CONVERT DECNUM (BASE 10) TO HEXNUM$ (BASE 16)
4020 IF DECNUM>255 THEN PRINT "***** DECNUM ERROR *****":CLOSE:GOTO 10030
4040 LONYBBLE=DECNUM AND 15
4060 HINYBBLE=(DECNUM AND 240)/16
4080 HEXNUM$=HEXVAL$(HINYBBLE)+HEXVAL$(LONYBBLE)
4100 RETURN
```

Listing 2 continued

Listing 2 continued

```
5000 '*************************************************************************
5010 ' UPDATE "ADDRL" AND "ADDRH" ADDRESS COUNTERS
5020 ADDRL=ADDRL+1
5040 IF ADDRL=256 THEN ADDRL=0:ADDRH=ADDRH+1
5060 RETURN

6000 '*************************************************************************
6010 ' CALCULATE LINE CHECKSUM AND WRITE TO HEX FILE
6020 CHKSUM=CHKSUM AND 255
6040 IF CHKSUM=0 THEN DECNUM=0 ELSE DECNUM=256-CHKSUM
6060 GOSUB 4000
6080 PRINT #2,HEXNUM$
6100 RETURN

9000 '*************************************************************************
9005 ' GET NEXT RECORD (BYTE) FROM OBJECT FILE
9010 GET 1,OBJPTR%:OBJPTR%=OBJPTR%+1
9020 RETURN
10000 '************************************************************************
10005 ' COME HERE FOR DATA READ ERRORS
10010 PRINT:PRINT "***** DATA READ ERROR *****"
10020 CLOSE

10030 SYSTEM
10040 END
```

End

Program Listing 3. Subroutine for Model I/III Basics.

```
8000 '*************************************************************************
8010 ' THIS SUBROUTINE SIMULATES THE "INSTR" FUNCTION
8020 SV=0
8030 FOR I%=LEN(OBJFILE$) TO 1 STEP -1
8040 IF MID$(OBJFILE$,I%,1)=SV$ THEN SV=I%
8050 NEXT I%
8060 RETURN
```

End

SAVE A BUNDLE

WITH ONE OF OUR BUNDLES

We have put together special bundles of software just in time to save you a bundle of money. Didn't you ever wonder why CP/M was so popular? The reason is the software availability. All programs (except EP) are for the Model 4/4P and have been optimized to install and run without hassle. For example, we have memory-mapped WordStar® and it runs circles around the standard version available elsewhere. We also added printer drivers for the Daisywheel II and the DMP-2100. Our CP/M® 2.2 is the best around. Read the reviews. You *know* the programs. You *know* the prices. This is the years' best value. You owe it to yourself to let these CP/M® programs and thousands of others (many in the public domain) start serving your needs.

Bundle #1
WordStar®
MailMerge®
SpellStar™
StarIndex™
Montezuma Micro CP/M®
$299 Delivered
RETAIL VALUE $1009

Bundle #5
Electric Pencil™
Red Pencil™
Blue Pencil™
$149 Delivered
Works with TRSDOS, NEWDOS-80, DOSPLUS and MULTIDOS (Model 1,3,4)
RETAIL VALUE $240

Bundle #2
WordStar®
DataStar™
ReportStar™
CalcStar™
Montezuma Micro CP/M®
$399 Delivered
RETAIL VALUE $1354

Bundle #3
WordStar®
MailMerge®
SpellStar™
StarIndex™
DataStar™
ReportStar™
CalcStar™
Montezuma Micro CP/M®
$499 Delivered
RETAIL VALUE $1699

Bundle #4
WordStar®
MailMerge™
SpellStar™
StarIndex™
dBASE II®
TurboPascal™
Montezuma Micro CP/M®
$599 Delivered
RETAIL VALUE $1553

Save A Bundle on these books and disks

Take advantage of our volume discounts. Buy any three items from this list and deduct $5 from your total order. Buy four...deduct $10. Buy five...deduct $15. Buy six...deduct $20 and so on. Buy a bunch, save a bundle. Please add $1 each for shipping. Add $5 to orders under $50 for handling.

TRS-80 Disk & Other Mysteries. The "How to" book of data recovery for the TRS-80 Model I disk operating system. 128 pages. Retail Retail $22.50 **Now $18**

Microsoft BASIC Decoded & Other Mysteries. The complete guide to Level II and BASIC. 312pages
Retail $29.95 **Now $24**

The Custom TRS-80 & Other Mysteries. The complete guide to customizing TRS-80 Hardware and Software. 336 pages
Retail $29.95 **Now $24**

BASIC Faster & Better & Other Mysteries. The complete guide to BASIC programming tricks and techniques. 290 pages
Retail $29.95 **Now $24**

BASIC Faster & Better Library Disk. Contains 121 functions, subroutines and user routines. Search, merge, compare and listing routines plus array handlers, BASIC overlays and video drivers.
Retail $19.95 **Now $18**

BASIC Faster & Better Demonstration Disk. Contains 32 demos of the Library Disk contents above
Retail $19.95 **Now $24**

BASIC Disk I/O Faster & Better & Other Mysteries. Programming tips and techniques to store/retrieve data from disk. 432 pages. Retail $29.95 **Now $24**

BASIC Disk I/O Faster & Better Demonstration Disk. All of the major demo programs and library of disk I/O subroutines in 25 BASIC programs. Random, Indexed Sequential and TREESAM file handlers included. Retail $29.95 **Now $24**

Machine Language Disk I/O & Other Mysteries. A guide to machine language disk I/O for the TRS-80. 288 pages
Retail $29.95 **Now $24**

TRSDOS 2.3 Decoded & Other Mysteries. Detailed explanation of the Model I disk operating system. 298 pages
Retail $29.95 **Now $24**

How To Do It On The TRS-80. A complete applications guide to the TRS-80 Model I, II, III, 100, and Color Computer. 352 pages
Retail $29.95 **Now $24**

The Custom Apple & Other Mysteries. Who cares?
Retail $29.95 **Now $19**

Due to the nature of this business, there are **NO REFUNDS** on software. We accept American Express, MasterCard, Visa and COD. Your card is not charged until we ship your order. Personal and company checks are accepted without delay (call for details). Delivered prices are for the lower nightmost 48 States only. Prices and specifications subject to change without notice. If you don't understand any of this, just give us a call and we will take the time to explain it until you do. First timers are always welcome here.

MONTEZUMA MICRO

214-339-5104
Redbird Airport, Hangar #8
P.O. Box 32027
Dallas, Texas 75232

"WE KEEP YOU RUNNING"

ORDER NOW . . . TOLL FREE
800-527-0347
800-442-1310
The Toll Free lines are for orders only
Specifications subject to change without notice

MONTEZUMA MICRO

P.O. Box 2169
Camp Varde (Lizard Flats)
Arizona 86322

"WE KEEP YOU RUNNING"

COMPUTERS

25 1000 16K Model 1000 128K RAM IBM Clone $849
25 1001 Model 1000 w 10M Hard Disk & 256K RAM 1699
26 1070 2DS dr Model 4 64K RS232 Pure P/S 1019
26 1127 64K Extended BASIC Color Computer 2 187
26 3134 16K Color Computer 2 99
26 3136 16K Extended BASIC Color Computer 2 125
26 1539 Pocket Computer 3 with 4K RAM 85
26 3650 Pocket Computer 4 51
26 3670 Pocket Scientific Computer 5 102
26 3802 Model 100 24K Portable Computer 539
26 3860 Model 200 24K Portable Computer 849
26-5103 Model 2000 2 drive 128K 1359
26 5104 Model 2000 W Hard Disk and 256K 2124
26 6021 Model 6000 2dr 512K 3824
26 6022 Model 6000 HD 512K 4674
26 6050 DT 1 Terminal 339
26 6052 DT 100 Terminal 675

MODEMS

25 1003 Model 1000 300 Baud Internal $ 127
25 1013 Model 1000 300/1200 Baud Internal 255
26 1084 Model 4P Modem Board 89
26 1173 DC Modem II 109
26 1174 Acoustic Coupler Modem 89
26 1176 DC 2212 300 1200 Baud 339
26 1177 DCM 5 300 Baud Modem 110
26 1178 DCM 3 300 Baud Modem 51
Anchor Mark I Modem 300 Baud with cable 74
Anchor 300/1200 Baud Modem with cable & PS 259
Hayes Smartmodem 300 Baud 159
Hayes 1200B 300 1200 Baud interna 439
Hayes Smartmodem 300 1200 Baud 479
Hayes Compatible 300 1200 uses same software 229

PERIPHERALS

ALSO SEE ADD ON CARDS UNDER UPGRADES

LNW-80 2dr cabinet with high drives 389
12" Green Non Glare Composite Video Monitor 89
12" Amber Non Glare Composite Video Mon for 85
12" Green Non Glare TTL (IBM) Video Monitor 99
12" Amber Non Glare TTL (IBM) Video Monitor 99
TTL Monitors come with IBM Cable at No Charge
Video cable for Model 1 keyboard to monitor 7
25 3010 Model 1200 VM 3 Monochrome Monitor 186
26 0511 Genuine RS Pantyhose Anti Glare Device 14
26 1085 Model 4P Carrying Case 21
26 1133 Model 3 15MB Hard Disk Inst Kit 85
26 1134 Model 4 15MB Hard Disk Inst Kit 68
26 1183 Model 100 Bar Code Reader 85
26 1185 Color Computer Koala Pad 51
26 1197 Mouse for Model 1000/1200/2000 85
26 1198 Serial to Parallel Converter 85
26 1208 CCR 81 Cassette Tape Recorder 51
26 1209 CCR 82 Cassette Tape Recorder 42
26 1248 Power Strip/Line Filter 42
26 1326 Anti Glare Panel Model 2/12 16 42
26 1342 Clear Keycover for Model 3/4 11
26 1410 Model 100 Modem Cable 17
26 1429 Automatic Power Strip/Surge Protector 59
26 1451 Non Automatic Power Strip/Filter 42
26 1457 Anti Glare Panel Model 1 3/4 42
26 2226 Color Computer Deluxe RS 232 Pkg 69
26 3008 Color Computer Joystick Pair Pair 17
26 3012 Color Computer Deluxe Joystick each 25
26 3024 Color Computer Multi Pak Interface 89
26 3025 Color Computer Mouse 43
26 3129 Color Computer HalfHigh Disk Drive 0 297
26 3130 Color Computer HalfHigh Add on Drive 169
Like above only parts It's better for 99
26 3211 Model 1000 VM 2 Monochrome Monitor 123
26 3212 Model 1000/1200 CM 2 Color Monitor 391
26 3503 PC 1 Cassette Interface 9
26 3505 PC 1 Carrying Case 13
26 3508 PC 2 Carrying Case 25
26 3612 PC 2 RS 232C Interface 36
26 3651 PC 4 Cassette Interface 33
26 3653 PC 4 1K Ram Module 16
26 3654 PC 4 Carrying Case 6
26 3804 Model 100 AC Adapter 5
26 3805 Model 100 Acoustic Coupler 39
26 3809 Model 100 Carrying Case 42
26 3813 Model 100 Soft Carrying Case 34
26 3812 Model 100 Legs (pair) 4
26 4154 Model 12 16B Internal 15Mb Hard Disk Kit 2245
26 4155 Model 2 3 4/12 16 15Mb Primary Hard Disk 1355
26 4156 Model 2 3 4/12 16 15Mb Secondary Drive 1100
26 4157 Model 2/12 16 Installation Kit for 26 4155 297
26 5111 Model 2000 VM 1 Monochrome Monitor 189
26 5112 Model 2000 CM 1 Color Monitor 509

FURNITURE

26 1324 Computer Table 68
26 1354 Computer Stand 42
26 1355 Printer Stand 42
26 1356 System Desk 102
26 1357 Printer Stand 85
26 1358 Corner for 1356 57 Desk Stand 34
26 1359 Hutch for 1358 Desk 51
26 1360 Printer Platform 25
26 4303 Deluxe System Desk 229
26 4305 Deluxe Printer Stand 128
26 4306 Terminal Stand 161
26 4307 Printer Stand 127
26 5115 Model 2000 Pedestal 76

PRINTERS & ACCESSORIES

26 1192 GCP 115 Color Graphics Printer 108
26 1196 GT 116 Graphics Pad 90
26 1255 DMP 120 120cps Dual Mode Printer 269
26 1261 TP 10 Thermal Matrix Printer 85
26 1268 CGP 220 Color Ink Jet Printer 509
26 1269 P1C 64 64K Printer Controller (Buffer) 212
26 1270 DWP 510 43cps Daisywheel printer 1270
26 1271 DMP 110 50cps Triple Mode Printer 225
26 1274 DMP 2100P 160cps Dual Mode Printer 1270
26 1275 TRP 100 Printer 255
26 1276 DMP 105 Creeep Printer 169
26 1277 DMP 430 Not So Cheeep Printer 765
26 1278 DWP 220 Replacement for DWP 210 509
26 1279 DMP 2200 High Speed Matrix Printer 1441
26 1280 DMP 130 100cps Triple Mode Printe 297
26 1441 Bi Directional Tractor for DMP 2100 102
26 1443 Bi Directional Tractor for DWP 210 109
26 1444 Bi Directional Tractor for DWP 220 102
26 1447 Bi Directional Tractor for Daisywheel II 188
26 1448 Single Bin Sheet Feeder for Daisywheel II 672
26 1455 Acoustic Cover 339
26 1459 Bi Directional Tractor for DWP-410 53
26 1461 Model 1/3 4/4P Printer Cable 29
26 1468 RS 232C Cable 16
26 1469 Model 100 Printer Cable 13
26 4401 Model 2 12 16 2000 Printer Cable 29
26 1490 10' RS 232C Cable 25
26 1491 25' RS 232C Cable 33
26 1492 50' RS 232C Cable 46
26 1493 100' RS 232C Cable 76
26 1495 RS 232C Cable Extender 17
26 1496 RS 232C Null Modem Adapter 15
26 1498 SW 302 Parallel Printer Switch 102
Try ours 2 position but only $9
26 1499 SW 303 RS 232C Selector Switch 128
Try ours 2 position but only $9
26 3551 PC 3 Printer 102
26 3605 PC 2 Printer 45
26 3652 PC 4 Printer 68

C.ITOH

GX 100 Gorilla Banana Serial or Parallel 50cps 139
ProWriter Jr 105cps NLQ Fric/Trac Parallel 229
3520 350cps Parallel IBM Compatible $ 1995
8510BP1 120cps Friction/Tractor Parallel IBM Comp 299
8510SEP Hot Dot 180cps NLQ IBM Graphics 399
8510SCP 7 Colors 180cps IBM Graphics 495
1550P Prowriter 2 120cps for Wide Paper Parallel 525
1550BCD Same as above except serial interface 299
1550SCP 7 colors 180cps Real Fast & Wide 695
1550SEP IBM Graphics 180cps NLQ 599
F10 40 Sprinwriter 40cps Daisywheel 899
F10 55 Printmaster 58cps Daisywheel FAST 999
F10 Bi Directional Tractor with Plastic Cover 199
F10 Electric Single Bin Sheet Feeder Inst Required 775
F10 Mechanical Single Bin Sheet Feeder 299
24LQ 24pin Dot Matrix 200cps 7 Colors 360cps 999
We have dust covers for most every printer we sell. Be sure to order one with your new printer to keep it like new.

CABLES

We manufacture a great number of different cable assemblies to connect most anything to anything else. Call us with your specific needs. We probably have it on the shelf. All at discount prices, of course.

SUPPLIES

5.25 SSDD Diskettes Pack of 10 1 Year Guarantee $ 14
5.25 DSDD Diskettes Pack of 10 1 Year Guarantee 18
8 SSDD Diskettes Pack of 10 1 Year Guarantee 29
8 DSDD Diskettes Pack of 10 1 Year Guarantee 34
5.25 Flipsort Holds 75 Disks 16
8.5 x 11 Tractor Paper 20lb 2900 Sheets 26
14 x 11 Tractor Paper 20lb 2900 Sheets 35
5.25 or 8 Head Cleaning Kit 9

RIBBONS

We have more ribbons than you do. Way too many types to list here. Add them to your order now. They're cheeeep too.

MONTEZUMA'S REVENGE

MODEL 3 UPGRADE
112K + CP/M + 80 COLUMN

The month Monte offers the deal of the year. The Holmes VID-80 is the only plug in printed circuit board which expands the Mod 3 display to 24 lines of 80 characters and allows operation of the CP/M 2.2 operating system with a 60K TPA. Both 80 x 24 and 64 x 16 modes are available from TRSDOS and BASIC. Reads and writes 18 different disk formats (Kaypro, Osborne, etc). A total of 112K of RAM. The extra 60K RAM can be set up as a RAM disk through software routines that are furnished with the supplied CP/M operating system. The VID-80 comes complete with easy to follow installation instructions and operational documentation. No trace cuts or soldering required. Guaranteed for one year parts and labor. Monty can ship this today.

MONTE'S SPECIAL PRICE .....$ 279
Add WordStar 3 3 complete and ready to run for $ 100

WE KEEP YOU RUNNING

SOFTWARE

Note: It is Radio Shack policy that not all multiple computer software packages contain disks for all models. Optional disks are available at slight extra charge. Be sure to ask for details when you order.

CP/M 2.2 by Montezuma Micro For The Model 4
Raised best by actual test! Get the good one $ 169
NEW! CP/M Hard Disk Driver with Backup & Restore 30
NEW! 25 3130 MS DOS & BASIC Disk/Doc Model 1200 76
26-0310 TRSDOS 2.3 For The Model 1 12
26-0312 TRSDOS 1.3 For The Model 3 12
26 0315 TRSDOS 6.2 Utility Disk 34
26 0316 TRSDOS 6.2 Disk+Doc+Ref Card 34
26 0413 Disk Drive Analyzer. Why pay more? 26
26 1507 Model 1/3 Stockpak 42
26 1510 Model 1/3 Terrapin 51
26 1511 Model 3 Home Accountant 85
26 1512 Model 4 Target PlannerCalc 85
26 1513 Model 1/3 Casa Spectacalculator 42
26 1514 Model 3 Portfolio Manager 169
26 1515 pfs file for Model 3 106
26 1516 pfs report for Model 3 85
26 1517 pfs report for Model 4 85
26 1518 pfs file for Model 4 107
26 1520 Model 4 VisiCalc 85
26 1521 Model 3 VisiCalc Business Forecast 85
26 1527 Model 3/4 Formation 169
26-1530 Model 4 Multiplan 169
26 1536 Model 4 Apartment Management 255
26 1539 Model 3/4 W-2 Writer 42
26 1540 Model 3/4 General Ledger 169
26 1541 Model 3/4 Accounts Receivable 169
26 1542 Model 3/4 Accounts Payable 169
26 1543 Model 3/4 Payroll 169
26 1544 Model 3/4 Invoice Writer 42
26 1545 Model 3/4 Inventory Control 169
26 1559 Model 1/3 Manufacturing Inventory Control 169
26 1560 Model 1/3 Fixed Assets 55
26 1562 Model 1/3 Profile 36
26 1563 Model 1/3 SCRIPSIT 85
26 1564 Model 1/3 Mailgram 14
26 1565 Model 1/3 Mail Merge 23
26 1588 Model 1/3 Medical Office Systems 254
26 1589 Model 3 VisiCalc Enhanced Version 85
26-1577 Model 1/3 Surveying 42
26 1579 pfs file for Real Estate 85
26 1580 Model 1/3 Property Manager 85
26-1581 Model 1/3 Personnel Manager 85
26 1582 Model 1/3 Time Manager 85
26 1584 Model 3 Checkwriter 80 85
26 1585 Model 3 Business Correspondence writer 127
26 1586 Model 1/3 Videotex Plus 42
26 1589 Model 3 MICRO Courier 127
26 1590 Model 1/3 SuperSCRIPSIT 169
26 1591 Model 1/3 Scripsit Dictionary 85
26 1592 Model 3 Profile Plus 169
26 1593 Model 3 Profile Plus LDOS/HD version 254
26 1594 Model 3 Desktop Plan 80 139
26 1595 SuperSCRIPSIT For The Model 4 169
26 1596 SCRIPSIT For The Model 4 85
26 1597 Model 3 Business Graphics Pak 146
26 1598 Model 4 Videotex Plus 42
26 1600 Dictionary for the Model 4 85
26 1608 **NEW!** Model 4 Desk mate 169
26 1630 TK! Solver for the Model 4 255
26 1635 Model 4 Profile 212
26 1922 Model 1/3 Orchestra 80 68
26 2011 Model 1/3 LDAS Tape Version 29
26 2012 3/4 Assembly Language Development Course 126
26 2013 LDAS Disk Version Model 1/3 29
26 2015 Model 3 Instruction Tapes 21
26 2017 Model 3 Assembly Language Course Tape 34
26 2018 Model 3 Assembly Language Course Disk 59
26 2019 TRSDOS 6 x Training Course 64
26 2022 Power Tool for the Model 1/3 43
26 2023 Dot Plot for the Model 1/3 17
26 2025 Constructor for the Model 4 42
26 2026 **NEW!** COBOL Query for the Model 4 127
26 2027 Graphics 80 for something 34
26 2180 Model 4 COBOL 169
26 2200 Model 3 FORTRAN 85
26 2201 Model 1 FORTRAN 85
26 2203 Model 1/3 COBOL 169
26 2205 Micro PILOT 34
26 2210 Model 3 BASCOM 169
26 2211 Model 3 Alcor PASCAL 212
26 2212 Alcor PASCAL For The Model 4 212
26 2213 Model 1 LDOS (LSI version) 69
26 2214 Model 3 LDOS (LSI version) 69
26 2216 CP/M Plus For The Model 4 127
26 GOOD The Good CP/M for the Model 4 Montezuma 169
26 REAL GOOD Hard Disk Driver for MM CP/M 30
26-EVEN BETTER Monte's Window for MM CP/M 49
26 EAR OUT Monte's Toolkit for MM CP/M 49
26 WOW Monte's BASCOM TRSDOS BASIC to CP/M 49
26 2217 CBASIC For The Model 4 (Requires CP/M 85
26 2218 BASCOM for the Model 4 165
26 2219 FORTRAN for the Model 4 85
26 2220-23 Videotex various applications 26
26 2224 Company Dow Jones Sign up Kit 17
26 2230 Model 4 C Language 212
26 2231 Model 4 Double Duty (128K req) 59
26 2718 Model 3 PILOT 101
26 2721 Color Computer LOGO Disk 85
26 2722 Color Computer LOGO ROM Pak 43
26 3030 OS 9 For The Color Computer 59
26 3034 **NEW!** OS 9 PASCAL for the Color Computer 85
26 3036 SO 9 BASIC For the Color Computer 85
26 3038 **NEW!** C Compiler for the Color Computer 85
26 3821 Model 100 Learning Lab 25
26 3829 Model 100 Multiplan ROM 127
26 3830 Model 100 Scripsit Cassette 34

ALL Color Computer GAMES 25% OFF Catalog Price
ALL PC(YDS) SOFTWARE 30% OFF Catalog Price

26-4501 General Ledger I 129
26 4502 Inventory Management System I 119
26 4503 Payroll 229
26 4504 Accounts Receivable 179
26 4506 Medical Office Systems 499
26 4511 VisiCalc 199
26 4512 Profile II 99
26-4513 Job Costing 99
26 4514 Order Entry 129
26 4515 Profile II Plus 199
26 4516 Profile Training Guide 58
26 4517 Profile Plus Upgrade 89
26 4520 Tyme Accounting 299
26 4531 SCRIPSIT 2.0 299
26-4532 SCRIPSIT Utility Disk 79
26 4534 SCRIPSIT Dictionary 129
26 4536 SCRIPSIT Printer Driver 39
26 4545 Litigation Support 199
26 4550 Business Graphics Analysis Pak 199
26 4554 Accounts Receivable 39
26 4555 Menu Generator 33
26-4556 Profile Forms 99
26 4557 Profile Archive 99
26-4558 Profile Prosort 116
26 4559 Prolook 139
26-4560 WESTLAW 159
26 4580 MultiPlan Model 2/12 212
26 4601 General Ledger 299
26 4602 Inventory Control System 199
26-4604 Accounts Receivable 299
26 4607 Order Entry ICS 299
26 4608 Sales Analysis 199
26 4621 Personnel Search 199
26 4701 FORTRAN 129
26 4702 EDAS 129
26-4703 COBOL 199
26 4705 Compiler BASIC 129
26 4707 COBOL Generator 499
26 4710 Program Editor 59
26-4712 Assembly Language Development System 199
26-4713 FDAS 1 39
26-4721 Videotex For The Model 2/12 16 42
26 4725 BASCOM BASIC Compiler 159
26 4742 CBASIC 159
26-4802 Inventory for the Hard Disk 299
26 4831 SCRIPSIT for the Hard Disk 299
26 4834 Dictionary for the Hard Disk 139
26 4835 SCRIPSIT for Printing Drives 263

ALL NEW MODEL 2000 GOODIES

26 5210 General Ledger 339
26-5212 Accounts Payable 339
26 5213 Accounts Receivable 339
26 5214 Inventory Control System 339
26 5280 Videotex Plus 42
26 5300 Lotus 1 2 3 399
26 5306 pfs report 106
26 5311 Multiplan 159
26 5330 Multiplie 199
26 5340 Home Accountant 106

MODEL 12 & 16 68000 SOFTWARE

26 6100 COBOL Development System 199
26 6201 General Ledger 449
26 6203 Payroll 499
26 6205 Accounts Payable 449
26 6209 Job Costing 149
26 6302 Inventory 299
26 6450 MAC16 Macro Assembler 299
26-6451 RM FORTRAN for Xenix 499
26 6455 COBOL for XENIX 499
26 6457 MBASIC for Xenix 499
Electric Pencil 2.0 Model 1/3 49
Electric Pencil for Model 2000 49
Blue Pencil Model 1/3 49
Red Pencil Model 1/3 39
Pencil Tip Printer Enhancer 29
Modem 80 A Powerful Communications Program M1/3 35
Trackcase for the Model 3 The original copy program 19
Super Utility Plus 3.2 Runs on Model 1/3 49
Super Utility Plus for Model 4 and 4P only 49
DoubleCROSS File transfer utility. Runs under MM CP/M. Allows unlimited file transfers both ways between any combination of TRSDOS 1.3, 6.x, PC MS DOS 1.x 2.x 3.x and 84 CP/M variations. Much faster than SuperCROSS XT or HYPER.
CROSSWORD includes Monte's Toolbox 49
SuperCROSS/XT Specify Model 1 3 or 4 79
CONV3TO4 Convert Mod 1/3 BASIC to Mod 4 39
CONV3TOPC Mod 1/3 BASIC to MS DOS by EMS 99
BASCON Converts Mod 4 BASIC to CP/M MBASIC 49
TURBO PASCAL Specify CP/M or MS DOS Version 3.0 59
TURBO PASCAL For MS DOS and 8087 84
TURBO PASCAL For MS DOS with 8087 and BCD 99
TURBO TUTOR Get started right. Disk & manual 29
TURBO TOOLBOX Complete PASCAL library with disk 39
TURBO GRAPHIX TOOLBOX IBM Hires Graphics 39
SIDEKICK The original disk organizer 34

CP/M OPERATING SYSTEMS

CP/M 2.2 for the Model 4 **The VERY BEST ONE** $ 169
Our Very Own Trick Hard Disk Driver 70
Puts Radio Shacks in The Shade 70
P&T CP/M 2.2m Floppy version Mode 2 12 16 180
P&T CP/M 2.2m for R/S Hard Disk Model 2 12 16 225

BOOKS and MANUALS

25 1501 MS DOS Reference Manual $ 29
25 1502 MS DOS BASIC Reference Manual 29
25 1503 Model 1000 Programmers Manual 12
25 1504 Model 1000 Technical Reference Manual 25
25 1505 Complete Guide to the Model 1000 13
26 1306 Introduction to MS DOS 7
26 1378 Newest Scriptebook Volume 7 7
26 2102 Model 1 Level II Manual 5
26 2111 Model 3 DOS Manual 6
26 2112 Model 3 BASIC Instruction Manual 5
26 2114 Sourcebook Volume 1 7
26-2115 Newsletter Book 1980 4
26 2119 Model 4 4P Technical Manual 33
26 2240 Newsletter Book 1981 6
26 2241 Newsletter Book 1982 9
26 3191 Color Computer Manual 5
26 3192 Extended Color Computer Manual 5
26 3197 Std Ext BASIC Manual for Color Computer 8
26 3810 Model 100 Technical Manual 8
26 3819 Model 100 Book by David Lien 13
26 4921 Model 2 Technical Reference Manual 26
26 4922 Model 12 Operating Manual 59
26 5403 Model 2000 Programmer's Manual 17
26 5404 Model 2000 Hardware Manual 21
26 6041 Model 16B Operating Manual 72
TRS-80 Disk & Other Mysteries by H. A. Pennington 19
Microsoft BASIC Decoded by Jonathan Valin 23
BASIC Faster & Better by Dave Hunter 23
BASIC Faster & Better Demonstration Disk 15
BASIC Faster & Better Library Disk 15
BASIC Disk I/O Faster & Better by Dave Hunter 23
BASIC Disk I/O Library Disk 28
The Custom TRS-80 by Dan Clements 24
TRSDOS 2.3 Decoded by Jonathan Valin 23
Machine Language Disk I/O by Mike Hughes 24
How Do It On The TRS-80 by W. Jeff Beckham 24
TRS-80 Beginners Guide 13
Using Super Utility + 3.x (all models) by Frank Hall 16

ADD-ONS & UPGRADES

16K RAM 200 nsec 8pcs 1 Full Year Guarantee $9
64K RAM 200 nsec 8pcs 1 Full Year Guarantee 29
256K RAM 8pcs 1 Full Year Guarantee 45
Model 4 128K RAM Upgrade Kit with Genuine PAL 58
25 1004 Model 1000 1st 256K RAM Upgrade Board 204
25 1005 Model 1000 2nd Disk Drive 169
Better than above (TEAC 55B) Save $$$$ 109
25 1006 Model 1000 RS 232 Board 85
25 1009 Model 1000 1200 512K RAM Board 212
25 1007 Model 1000 Hard Disk Controller Bd 255
25 1009 Model 1000 2nd 256K RAM Upgrade Board 169
25 1010 Model 1000 Mouse & Clock Calendar 85
25 1011 Model 1000 Memory Plus Expansion Board 272
25 1014 Model 1000 Plus RS 232 Interface 81
25 1015 Model 1000 Plus Mouse Calender Board 85
25 1025 Model 1000 10M Internal Hard Disk 595
Our own 10M internal Hard Disk complete with controller cable brackets and instructions 499
26 3040 Model 1200 Monochrome Adapter 186
Same except ours Save $$$$ 99
26 3043 Model 1200 Graphics Display Adapter 264
Same except ours Save $$$$ 113
25 3044 Model 1200 Technical Graphics Master 499
26 1125 Model 3 Hi Resolution Graphics Board 139
26 1126 Model 4 Hi Resolution Graphics Board 225
26 1127 62 Model 3 4 Drive 0 Kit 399
Same as above except our brand Save $$$$ 299
Famous AEROCOMP DDC Double Density Controller 99
26 1145 Model 1 RS232 Board for Model 4 89
Model 3 4 RS232 installation kit less cable 69
26 1163 Radio Shack 40 Track Disk Drive 149
Same as above except 40 track Tandon Save $$$ 119
MODEL 4 5Mhz Speedup Mod Add $10 for 4P 69
Model 4 256K RAM Board Expands to 1 Megabyte 199
26 4104 Model 2 Hi Resolution Graphics Board 424
26 4105 64K RAM Board 339
26 4167 8 Slimline Double side Drive 339
26 5140 Model 2000 Hi Resolution Graphics Board 255
26 5141 Model 2000 Color Graphic Chip Set 106
26 5144 Model 2000 Mouse Clock Board 85
26 5160 Model 2000 Internal 128K Board 153
26 5161 Model 2000 External 128K Board 187
26 5162 Model 2000 128K RAM for 26 5161 110
26 6010 Model 2 or 12 to Model 16 68000 Kit 699
26 6011 Model 16 128K RAM Board 299
26 6012 Model 12 128K RAM Kit 99
26 6013 Model 16 3 User RS 232 Card 159
26 6016 Model 16 512K RAM Kit 725
Same except ours Save $$$$ 330
26 6017 Model 12 Card Cage 159

BUY FROM US RIGHT NOW!

(Please read all this fine print first)

Our inventory is so large it can not be listed completely. Please call if you do not see what you want. Chances are we have it. Because of the time lag in magazine advertising prices are subject to change without notice and are mail order only. We are not responsible for typographical errors. Your company and/or personal checks are welcome and will not delay your order. We use TeleCheck. Please observe the following requirements and your order will be shipped without delay. The check must be drawn on a USA or Canadian bank and payable in US Dollars. It must be a bank printed check and contain your street address (not PO Box or APO/FPO) and telephone number. The signature must be EXACTLY match the name printed on the check. That's all there is to it. In addition we accept American Express, Mastercard, Visa, Cashier's Checks, Wire Transfer, and we also ship COD. COD's and motor freight shipments may require a deposit. Some special items require deposits. ALL COD's require cash or cashier's check in advance. Company and personal checks can NOT be accepted on payment of COD shipments. Credit cards are not charged until we ship your order. **SHIPPING CHARGES ARE NOT INCLUDED.** Please add $5 handling to all orders totaling less than $50. Add $20 handling charge to all orders over $500 shipped to Canadian addresses. No State Sales Tax collected on out of state shipments. If you order by 5PM your order will be shipped the NEXT DAY if stock is on hand. Responsibility of suitability of software rests with the purchaser. Due to the nature of this business there are **NO REFUNDS ON SOFTWARE. REPLACEMENTS WILL BE PROVIDED FOR DEFECTIVE SOFTWARE ITEMS** provided we are notified within ten days of delivery of the merchandise. If you have a particular problem please call us for help and instructions.

WARRANTY

All items carry the original manufacturers warranty. We will be happy to mail you a copy upon request.

CALL TOLL FREE FROM ANYWHERE IN THE USA

800-527-0347
800-442-1310

VISA MasterCard AMERICAN EXPRESS

Bringing GW-Basic Up to Speed

When *80 Micro* techie Beve Woodbury converted one of this month's graphics programs (Sinewave, "Window Screens," p. 58, Program Listing 1 and Photo 1) from Model 4 BasicG to Model 1000 GW-Basic, she witnessed a profound drop in speed. The 1000's Basic interpreter, like a bureaucracy, is large, complex, and sl-o-o-w. So I wrote the machine-language subroutine in Program Listing 1 that speeds things up. Sinewave (Program Listing 2) still won't move on the 1000 like it does on the 4, but at least it's in color.

My explorations led me to three areas I'd like to discuss: using the Call statement, memory-mapping graphics, and using Debug with Basic.

The Sinewave program displays a series of overlapping framed windows (via the View statement) that rise and fall sinusoidally and give you the illusion of three dimensions. The chief laggard is the window-clearing operation. The machine-language subroutine clears them instantly, using the X and Y coordinates of the upper left- and lower right-hand window corners passed as integer variables in the Call statement. (If you want to see how sluggish Model 1000 Basic graphics really are, replace the Call statement in line 110 of Listing 2 with a simple CLS.)

The program runs on a 128K Tandy 1000 even though graphics mode 6 (640 columns by 200 rows, four colors) takes 32K. This mode limits it to the 1000 and PCjr. You can modify Sinewave's machine-language section to run in screen mode 2 on an IBM or 1200.

Basic Points

Here's how the Model 1000 version of Sinewave works. The Clear statement in line 10 limits Basic data space to 8,192 bytes (2000 hexadecimal [hex]), enough for this small program. Clear also sets

System Requirements

Model 1000
128K RAM
GW-Basic

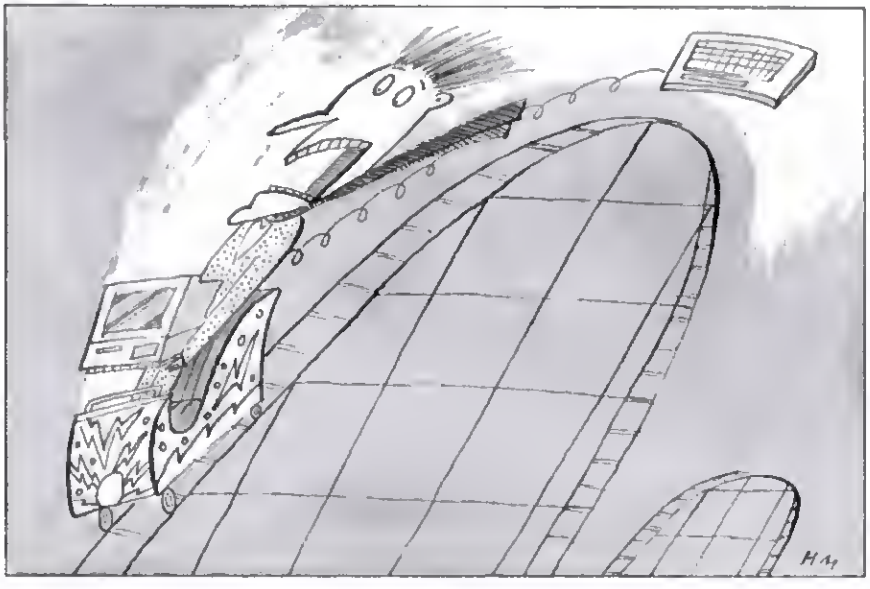


Illustration by Hal Mayforth

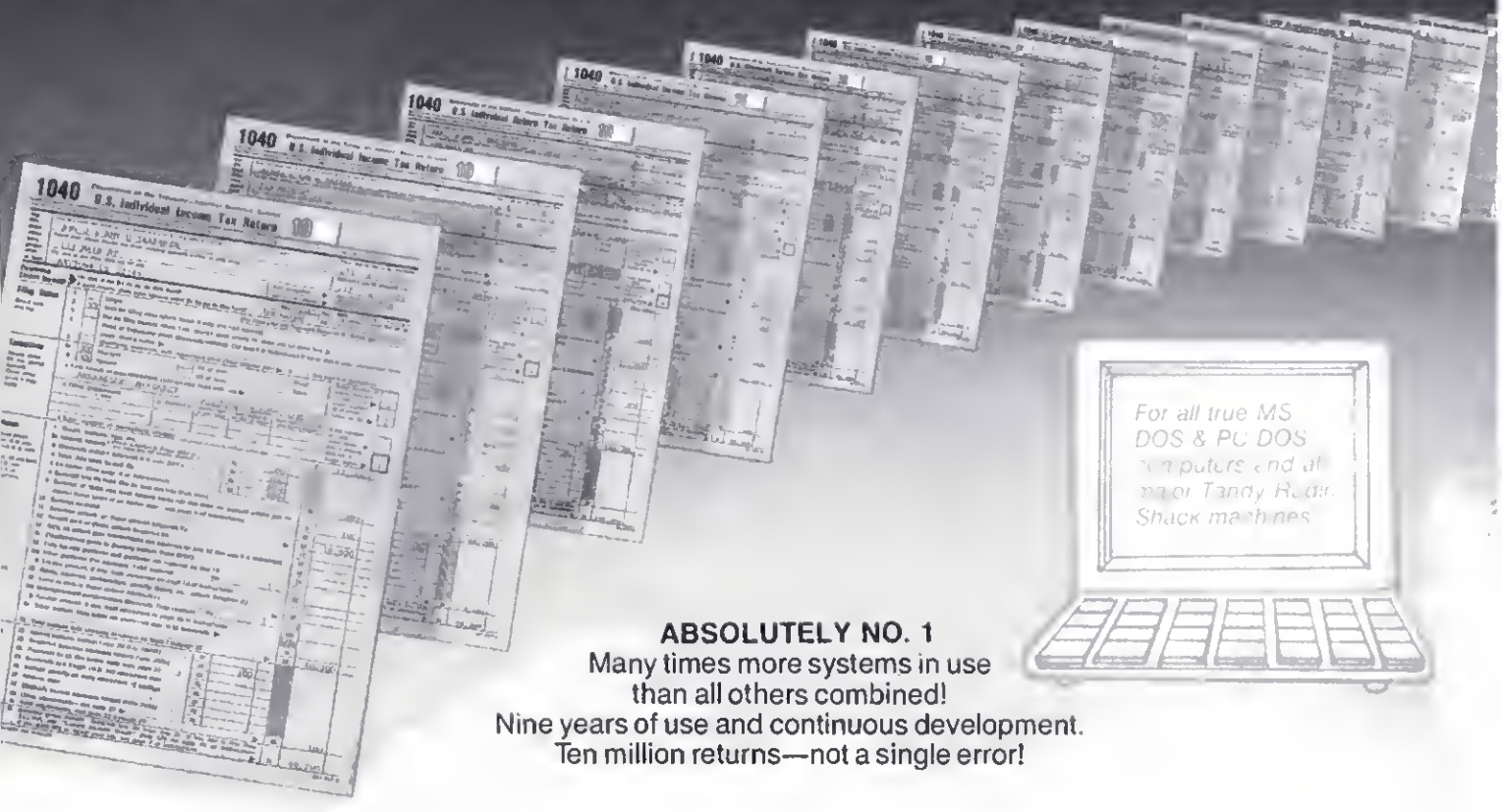
Program Listing 2. Assembly-language source code for Sinewave subroutine.

```
10 CLEAR ,&H2000,,32768! : SCREEN 0 : KEY ON : KEY OFF
15 SCREEN 6 : DEFINT I-N : I=&H2000
20 FOR J=I TO I+230 : READ K : POKE J,K :NEXT J
30 C=55 'the lower the number, the flatter the wave
40 J=0:Z1=5:Z2=.9:A=1:B=12
50 FOR X=A TO 8 STEP .15
60 X1=20*X:Y=SIN(X):Y1=119-C*(Y+1)
70 IF C<0 THEN PRINT "TROUBLE   C<0":END
80 IF J>0 THEN C=C-.04
90 IF J>0 THEN X1=X1+2:Y1=Y1-.01
100 IF X1<0 TREN X1=X1* -.1
105 IX1=X1:IY1=Y1:IX2=X1+Z1:IY2=Y1+Z1
110 VIEW(X1,Y1)-(X1+Z1,Y1+Z1),,1:CALL I(IX1,IY1,IX2,IY2)
120 Z1=Z1+Z2:NEXT
130 Z2=-Z2:J=J+1
140 IF J=2 THEN 160
150 A=12:B=23:GOTO 50
160 IF INKEY$="" THEN 160
1000 DATA 235,10,0,0,0,0,0,0,0,0,0,0
1010 DATA 85,139,236,6,191,2,32,139,118,12,184,127
1020 DATA 2,232,170,0,139,118,10,184,199,0,232,161
1030 DATA 0,139,118,8,184,127,2,232,152,0,139,118
1040 DATA 6,184,199,0,232,143,0,161,2,32,186,0
1050 DATA 255,232,155,0,163,10,32,137,22,2,32,161
1060 DATA 6,32,186,127,0,232,139,0,137,22,6,32
1070 DATA 43,6,10,32,124,103,72,139,240,161,4,32
1080 DATA 177,4,246,241,139,216,130,196,152,185,0,32
1090 DATA 247,225,80,138,195,177,160,246,225,91,3,195
1100 DATA 139,216,161,8,32,43,6,4,32,124,62,64
1110 DATA 139,200,184,0,184,142,192,252,139,251,3,62
1120 DATA 10,32,3,62,10,32,161,2,32,38,33,5
1130 DATA 71,71,81,139,206,184,0,0,243,171,89,161
1140 DATA 6,32,38,33,5,129,195,0,32,129,251,63
1150 DATA 127,118,8,129,235,0,128,129,195,160,0,226
1160 DATA 203,7,93,202,8,0,139,20,131,250,0,125
1170 DATA 3,186,0,0,59,208,126,2,139,208,137,21
1180 DATA 71,71,195,179,8,246,243,138,204,152,211,234
1190 DATA 138,242,195
```

End

THE ULTIMATE SOLUTION TO TAX PREPARATION

PC-Tax™ for TaxPros



ABSOLUTELY NO. 1
Many times more systems in use than all others combined!
Nine years of use and continuous development.
Ten million returns—not a single error!

PCTAX DOES:

Give the serious Tax Preparer the finest, time-tested, most sophisticated Tax Preparation System ever devised.

Supplies unlimited telephone support through the Tax Season.

Ships systems to your order far in advance of the Tax Season, giving you time to familiarize and hone your skills.

TO GET FACTUAL: Computes all tax amounts, all fixed & income-related limitations. Automatically totals W-2's, income averages, computes depreciation, etc., etc.

Does returns twenty times faster than anyone can do them manually and ten times faster (100 times easier) than any other "Tax System," 100 times faster than sending them out to a service bureau.

PCTAX DOES NOT:

Offer a "hot-shot amateur" or two- or three-times failed system. (To the shame of the software field and the total exasperation of accountants, there were over 68 of these last season.) We hate it, but there it is.

Doesn't require you to buy some oddball "operating system." You put our disk in your machine and it does taxes. That's all!

Doesn't bug you or use up disk space with rainbow-colored "bouncing-ball" displays. We figure you already have a TV.

If you do income tax returns, you need PCTax, the truly professional, computerized income tax preparation system. Look into it today.

Contract Services Associates, 507 Lead, Kingman, Arizona 86401, (602) 753-1133.

CSA CONTRACT SERVICES ASSOCIATES

For *FINAL* solutions

Circle 532 on Reader Service card.

Program Listing 1. Sinewave Basic program for the Tandy 1000.

```
; WINDOW clears a rectangular area of the screen (to background)
; in Basic screen mode 6 (high res, 4 colors). The x and y
; coordinates of the upper-left and lower-right corners are
; passed in the Call statement (IX1%,IY1%,IX2%,IY2%).
;-----------------------------------------------------------------

code segment

window    proc far
     public    window
     assume    cs:code, ds:code
     org  2000H          ;start at 2000 Hex within Basic's space
start:
     jmp  short pastdata

; data
x1        dw   ?
y1        dw   ?
x2        dw   ?
y2        dw   ?
leftedge  dw   ?

pastdata:
     push bp        ;save Basic's BP, then use it to
     mov  bp,sp     ;point to passed variables on stack
     push es        ;segment registers must be restored

; get variables from stack, check bounds, and store
     mov  di,offset x1   ;bx points to location of x1 storage
     mov  si,[bp+12]     ;location of x1 on stack
     mov  ax,639
     call bounds
     mov  si,[bp+10]     ;location of y1
     mov  ax,199
     call bounds
     mov  si,[bp+8]      ;location of x2
     mov  ax,639
     call bounds
     mov  si,[bp+6]      ;location of y2
     mov  ax,199
     call bounds

; determine left and right byte masks, store word position within
; line and count for columns in row (-2)
     mov  ax,x1          ;get left margin for division
     mov  dx,0ff00H      ;premask for left edge
     call wmask          ;determines mask word
     mov  leftedge,ax    ;number of word within line (0-79)
     mov  x1,dx          ;store mask in x1
     mov  ax,x2          ;get right margin for division
     mov  dx,007fH       ;premask for right edge
     call wmask
     mov  x2,dx          ;store mask in x2
     sub  ax,leftedge    ;subtract left word from right
     jl   leave          ;if x1>x2 then get out
     dec  ax             ;fudge
     mov  si,ax          ;si stores col. count

; determine memory location of first row (in section 1,2,3 or 4) and
; number of rows
     mov  ax,y1          ;get top y dimension
     mov  cl,4
     div  cl
     mov  bx,ax          ;temporary storage of results
     mov  al,ah
     cbw                 ;ax has word remainder
     mov  cx,2000H
     mul  cx             ;ax has video block memory location
     pusb ax             ;save it
     mov  al,bl          ;quotient in al
     mov  cl,160
     mul  cl             ;ax has offset in video block
     pop  bx             ;put vid block addr in bx
     add  ax,bx          ;ax has start video line address
     mov  bx,ax          ;keep in di
     mov  ax,y2          ;get lower bound
     sub  ax,y1          ;subtract upper bound
     jl   leave          ;if y1>y2 then abort mission
     inc  ax             ;fudge
     mov  cx,ax          ;use number of lines as count

; point ES to video memory (B800H) and set up loop parameters
     mov  ax,0B800H      ;start of video memory
     mov  es,ax          ;es points to it
     cld                 ;inner loop (string move) increments

loop1:                   ;outer loop - set row; do left edge
```

Listing 2 continued

aside the 32,768 bytes of high RAM needed for one screen of high-resolution, four-color graphics. Above Basic's reduced work area, and below the overgrown video RAM, is room for the machine-code subroutine, even with a 128K 1000. Changing from screen mode zero to 6 clears the screen rapidly. Waiting for the screen to clear in mode 6 induces sleep.

The DEFINT statement in line 15 and the variable assignments in line 105 ensure that the subroutine receives the window corner coordinates as integers (much easier to deal with). Line 20 POKEs the 231 bytes of 8088 machine code (lines 1000–1190) into memory, starting at offset 2000 hex in Basic's data area (protected by the Clear statement). The Call statement (line 110) sends execution to that memory offset, stored in variable I, and pushes the locations of the four passed variables onto the stack. The Call offset must be a variable. The subroutine replaces the sluggish CLS statement.

You can use two methods to reserve memory for machine-language subroutines in Basic. You can use the /M: parameter to make space for your subroutine above Basic. It's invoked when loading Basic, and controls the size of Basic's data area—the default is the maximum of 64K. Or you can use the Clear statement to reserve space within Basic's data area. There's an important difference; a machine-language subroutine loaded above Basic isn't protected from a "child" process called by the Basic Shell command. If you use Shell to load Debug above Basic, it'll load over any code Basic has put there. Use both if you want to shrink Basic's work space and protect your subroutine from a child process.

Subroutine Source

I used the Tandy 2000 version of MASM to assemble the source code on my 1000 and the MS-DOS linker to create an EXE file. Use the MASM assembly listing to get the actual code for the Basic Data statements. You can list it in hex format, e.g., &HFF. I converted hex to decimal for easier typing.

When accessed by a Basic Call, your subroutine should first set up the BP register to point to the passed variable locations on the stack. The Basic manual explains this process. Remember that the values stored on the stack are not the variables themselves, but their offset in Basic's data segment. The locations are on the stack last in/first out, but above the 4-byte return address and the 2-byte BP register you've pushed onto the stack. The far return that ends the subroutine must throw off the number of

Continued on p. 98

Circle 363 on Reader Service card.

Price Changes Occur On A Daily Basis. Please Call 1-800-343-8841

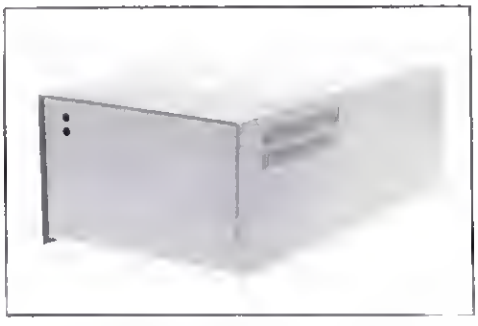
PRICE BREAKTHROUGH

Price Changes Occur On A Daily Basis. Please Call 1-800-343-8841

80

MEGA HOLIDAY SPECIAL CALL US!

Super Sale on New Hard Drives



Introducing

MEGADISK™

Winchester Hard Drive Ready to run on the TRS 80 Model I/III/IV/4P, Color Computer, I.B.M.-PC, Max/80. Software Drivers: LDOS, NEWDOS/80, DOSPLUS, TRSDOS 6.x
*Montezuma Micro CP/M available

Fully Warranteed

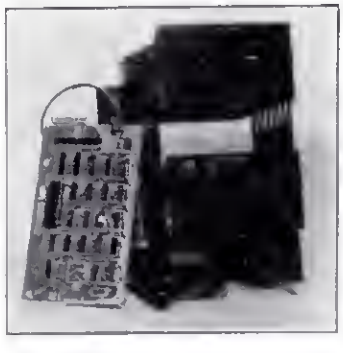
DRIVE A HARD BARGAIN™ starting at $499.95

MEGAPLEX your Megadisk for $499.95

Use up to 10 computers, multiplexed with 1 megadisk

For the TRS-80 models I, III, IV, 4P and Max/80

Call Toll Free Ordering 1-800-343-8841



$269.95 SPECIAL

Disk Drive Upgrade Kit
for Model III/IV easy to install system — no soldering.
Complete with controller, towers, power supply, 1 disk drive, cables, and easy to follow instructions.
Starting at $269.95
Second Drive $89.95

CANADIAN CUSTOMERS PLEASE CALL 514-383-5293

80

© 1985 SOFTWARE SUPPORT, INC.

1 Edgell Road, Framingham, MA 01701 (617) 872-9090 Telex-383425

Hours: Mon. thru Fri. 9:30 am to 5:30 pm (E.S.T.) Sat. 10 am to 3:30 pm

SERVICE POLICY — Our Professional Technical Staff Is Available To Assist You Monday Through Saturday.
WARRANTIES — Up To One Full Year Parts And Labor. Floppy Disk Drive Power Supplies — Five (5) Years.
SERVICE — 24 Hour Turn-A-Round On All In-Stock Parts. Dealer Inquiries Invited. Call 617-872-9090

Please Call For Shipping, Handling And Insurance.
Cash Discount Prices

Toll Free 1-800-343-8841

Please Call For Our Latest Price Saving Specials.

Not Responsible for Typographical Errors. Prices and Specifications May Change Without Notice.

Prices Change Every Day. Please Call 1-800-343-8841 For Lower Prices.

PRICE BREAKTHROUGH

DEALER INQUIRIES INVITED

Prices Change Every Day. Please Call 1-800-343-8841 For Lower Prices.

MEGADISK™ HARD DISK DRIVE SYSTEMS — **TOLL FREE ORDERING 1-800-343-8841**

For the { IBM/PC, Tandy 1000, TRS/80 Models I/III/IV/4P, Compaq, Tava, PC Workalikes, Color Computers, Heath/Zenith, Max/80
Complete with Hardware, Cables, Software and Quikfit Installation

5 Megabytes Internal Mount IBM/PC	starting at $	279.95
10 Megabytes Internal Mount IBM/Tandy 1000	starting at	439.95
20 Megabytes Internal Mount IBM/Tandy 1000	starting at	579.95
5 Megabytes External System	starting at	499.95
10 Megabytes External System	starting at	749.95
20 Megabytes External System	starting at	899.95
Tape Backup System — Internal Or External (IBM/PC)	starting at	449.95

Wow! New Low Prices

DOS Systems Available: { IBM/Heath — DOS, 1.0, 2.0, 2.1, 3.0, or later
TRS/80-LDOS, TRSDOS 6.x, Newdos/80, Dosplus, CP/M, COCO DOS, Max/80 LDOS, OS9

FULLY WARRANTIED — PARTS AND LABOR — 24 HOUR SERVICE — CALL TOLL FREE — 1-800-343-8841

FLOPPY DISK DRIVES, POWER SUPPLIES AND CABINETS

Our Disk Drives are UL approved — Our Floppy Drive Cabinets and Power Supplies are Underwriters Laboratory Listed and have passed the required Federal Communications Part 15 Section B-EMI/RFI tests.
Warranty on all disk drives is one full year parts and labor. Warranty on floppy disk drive power supplies is five (5) years. In warranty or out of warranty service is 24 hour turn-a-round on all disk drives and power supplies.

Full Height — Tandon		
100-1	Single Sided 40 tk Bare	$ 99.95
	In Case with Power Supply	139.95
	Dual Drives in One Cabinet	239.95
100-2	Dual Sided 40 tk Bare	109.95
	In Case with Power Supply	149.95
	Dual Drives in One Cabinet	259.95
Half High Drives — Tandon/TEAC		
	Single Sided 40 tk Bare	79.95
	In Case with Power Supply	119.95
	Dual Drives in One Cabinet	209.95
	Dual Sided 40 tk Bare	**109.05**
	In Case with Power Supply	**149.95**
	Dual Drives in One Cabinet	**259.95**
Apple/Franklin Disk Drives		
	35/40 Track in Case with Cable and Software	129.95
Commodore Disk Drives		236.95
Power Supplies and Cabinets 5¼" and Hard Drive Systems		starting at 42.00

COLOR COMPUTER DISK DRIVE SYSTEMS AND ADD IN PRODUCTS

40 Track Single Head Drive with Case, Power Supply, Cable Controller, Instruction Booklet, Diskettes	Special $ call
Above with Dual Drives in One Cabinet	289.95
40 Track Dual Head with Case, Power Supply, Cable, Controller, Instruction Booklet, Diskettes	**259.95**
Above with Dual Drives in One Cabinet	**379.95**
Dual DOS Switch	29.95
With Second DOS System — JDOS, RSDOS, and Booklet	69.95

MODEMS

Volksmodem 300 Baud	$ 69.95
Signalman Mark X Autodial	123.95
Mark XII 1200/300 Baud Autodial	284.95

ALL IN-STOCK ITEMS SHIPPED WITHIN 24 HOURS. SAME DAY SHIPPING PROVIDED BY REQUEST WITHOUT ANY EXTRA HANDLING CHARGES.

********** MEGADISK SPECIAL **********

Megaplex your Megadisk. Use up to 10 TRS-80 Model I, III, IV, 4P, Max/80 computers with one hard disk. Prices starting at $499.95

IBM - PC/XT WORKALIKE

All of our computers have: 8 slot motherboard, 640K, monochrome adapter, parallel printer port, 130 watt power supply, free software, monitor, serial port, clock calendar — all with our full warranty.

/PC- 2 Floppy Drives - Monochrome Monitor, MS-DOS	$1,299.00
5meg/XTtra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	$1,599.00
10meg/XTtra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	1,799.00
20meg/XTtra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	1,995.00
Internal Tape Backup For Any Of Above systems Add	$449.95
Color for above systems	500.00

PRINTERS

Dot Matrix	
Citizen	$ Call
Star Micronics — S.G. Series	starting at $259.95
Panasonic 1090	249.95
Daisy Wheel	
Silver Reed 440 80 Column 12 CPS	315.95
550 132 Column 19 CPS	439.95
770 132 Column 36 CPS	895.00
Olympia 132 Column 14 CPS with Form and Tractor Feed	399.95
Apple/Franklin Printer Interface w/Graphics and Cable	84.95
Printer Cables	starting at 19.95
Printer Paper — Microperf Edge 1000 Sheets	16.95

ELECTRICAL

Surge Protectors — Line Filters — SL Waber — 6 Outlets with Switch	$ 39.95
Uninterruptable Power Supplies	399.95

MISCELLANEOUS

Diskettes in 10 Pack	**from $ 9.95**
Twoprint Switches	from 99.95
Disk Drive Cables	from 16.00
Maintenance Cleaning Kits	12.00
Parallel Printer Buffers 8K	149.95
Floppy Disk Drive Cables	
1 Drive	16.00
2 Drives	18.95
Heath/Zenith 2 Drive Cables — Shielded	24.95

80 © 1985 **SOFTWARE SUPPORT, INC.**

1 Edgell Road, Framingham, MA 01701 (617) 872-9090 Telex-383425

Hours: Mon. thru Fri. 9:30 am to 5:30 pm (E.S.T.) Sat. 10 am to 3:30 pm

SERVICE POLICY — Our Professional Technical Staff Is Available To Assist You Monday Through Saturday.
WARRANTIES — Up To One Full Year Parts And Labor. Floppy Disk Drive Power Supplies — Five (5) Years.
SERVICE — 24 Hour Turn-A-Round On All In-Stock Parts. Dealer Inquiries Invited. Call 617-872-9090

Please Call For Shipping, Handling And Insurance. Cash Discount Prices

Toll Free 1-800-343-8841

Please Call For Our Latest Price Saving Specials.

Not Responsible for Typographical Errors. Prices and Specifications May Change Without Notice.

Listing 2 continued

```
        mov  di,bx          ;start of row
        add  di,leftedge    ;start in row in words, but
        add  di,leftedge    ;must be in bytes
        mov  ax,x1          ;get leftmask
        and  es:[di],ax     ;do left edge of row
        inc  di             ;point to next column
        inc  di             ;which is next word

; inner loop - print row (center bytes if any)
        push cx             ;save outer loop counter
        mov  cx,si          ;number of inner columns
        mov  ax,0           ;things will be black
rep     stosw               ;shove those words
        pop  cx             ;recover outer loop counter

; display right byte (leave outside of rectangle untouched)
        mov  ax,x2          ;get right mask
        and  es:[di],ax     ;do right edge

; adjust for next row
        add  bx,2000H       ;point to next video block
        cmp  bx,7f3fH       ;is it above video memory
        jbe  continue       ;if not then cont.
        sub  bx,8000H       ;if yes then put it in lower block
        add  bx,160         ;and point to next row
continue:
        loop loop1
leave:
        pop  es         ;restore registers for Basic
        pop  bp
        ret  8          ;discard 4 passed words and return (far)

window  endp
;------------
; near subroutine to check bounds of passed variable and store it

bounds  proc near
        mov  dx,[si]        ;si points to Basic variable
        cmp  dx,0           ;is variable greater than 0?
        jge  pos
        mov  dx,0           ;if not, than make it 0
pos:    cmp  dx,ax          ;does variable exceed limit?
        jle  less
        mov  dx,ax          ;if greater than set at limit
less:
        mov  [di],dx        ;store variable
        inc  di             ;point to next storage area
        inc  di
        ret
bounds  endp
;------------
; near subroutine to determine mask word for left or right edge

wmask   proc near       ;premask in dx, x-coord. in ax
        mov  bl,8
        div  bl             ;divide x-coord. by 8
        mov  cl,ah          ;put remainder in counter
        cbw                 ;ax has quotient (word in line)
        shr  dx,cl          ;right byte of ax is mask
        mov  dh,dl          ;both bytes of mask are the same
        ret
wmask   endp
;------------
code ends
        end  start
```

End

bytes used to pass the variable locations—RET 8 in this case—or your computer will hang up.

Storing data in the subroutine code, as I have, creates complications. The machine-language instructions are all position-independent; the data is not. The ORG 2000H directive provides that data offsets correspond to where they're POKEd in Basic's data segment (starting at 2000 hex). When the machine code requests the word stored at offset 2002 hex, it'll be there.

I also put DS in the Assume directive so the assembler doesn't add a CS: prefix to every data reference because an extra byte for each reference adds up. The first Jump instruction (past the data) just makes it easier to call the subroutine; the first instruction is the entry point. I could have put the data at the end.

The interfacing approach I took, storing the subroutine in Basic's data area, is the most flexible when you want to run a program on differently configured machines. Because the subroutine's loaded relative to Basic's data area, it isn't set at any specific memory location. If you use only one memory configuration, and know where Basic loads (see below), you can put your code at a specific memory location above Basic by POKEing data in a loop, or by BLOADing a binary file. In either case, you must first change the CS register (with DEF SEG) to point to the desired memory location. Remember that the value in a DEF SEG statement is a segment address, i.e., the actual address divided by 16.

If you load your subroutine at a set address, and have a data area in your subroutine, you can usually save some bytes by using DS to reference the data locations. As in Listing 1, include DS in the Assume directive so the assembler doesn't add a CS: override to each reference. Because you're not using Basic's data area in this case, you must load the DS register with the contents of CS (after saving DS, of course). But remember that the variable location offsets passed in the stack are in Basic's data segment. You can use an ES override prefix to get these values after loading ES with the Basic data segment. You must restore all segment registers other than CS before return.

Beware of the Tandy 1000 Basic manual on this point. It's sprinkled with statements that DEF SEG alters the DS register. This just isn't true; only the CS register changes. If you disregard this rather fundamental error and consult the IBM manual, the Tandy manual is quite helpful.

Memory Mapped

I used direct video memory addressing to clear the desired screen area. BIOS calls that set pixels, though easier to code, aren't fast enough. The screen modes and addressing are exactly like the PCjr's. Location of video memory in RAM depends on memory size, but you can always address it through a 32K window beginning at memory location B8000 hex (segment B800). You pay no time penalty for addressing video memory through this window, and it's always at the same location. I used the String Store command (STOSW) to move zeros quickly to areas of video memory, blacking them out. Therefore, I set the destination segment register (ES) to B800 hex, the video window.

I chose the most complicated graphics mode to get high-resolution and color, too. It takes 2 bits per pixel to code for four colors, but the two pixels are in different bytes of video RAM. Every 2 consecutive bytes code for eight pixels with corresponding bits in the 2 bytes coding the color of one pixel.

If the left-most bit (7) of hex bytes B800:0000 and 0001 is set, the pixel in the upper left screen corner is white (default palette). If both bits are zero, the upper left pixel is black. Combinations of set and unset bits produce cyan and magenta pixels. Bit 6 of those 2 bytes codes for the next pixel in the top row.

Circle 352 on Reader Service card.

Circle 437 on Reader Service card.

Circle 95 on Reader Service card.

Because the 8088 CPU deals in bytes and words, complicated graphics involves lots of bit manipulation.

To further complicate things, the 160-byte rows are not arranged contiguously in memory. The 32K video space is divided into 8K sections, every section containing every fourth screen row, but not the same rows as any other section. B800:0000–1F3F contains rows zero, 4, 8, 12, and so on through 196. The next section has rows 1, 5, 9, 13, and so on through 197.

I used 2-byte masks to And the left and right edges of the window being cleared. Both bytes in a mask are the same, and zeros in the mask correspond to pixels to be blacked out inside the window. The area in a row between the masked edges comprises whole words representing eight-pixel groups. You can quickly clear these by loading the corresponding words with zeros via a String command. Most of the program determines which row to start on and how many rows, which word in each row is the left edge and how many words to the right edge, and what masks to use on the left and right edges.

Debugging from Basic

Debugging Basic machine-language subroutines on the 1000 isn't easy. GW-Basic is an EXE file and can load anywhere in memory (but always in the same place under given conditions). You can find Basic's data segment from Basic by executing the following line right after loading it:

```
PRINT HEX$(PEEK(&H04A6)):HEX$(PEEK
      (&H04A5))
```

Basic will use the 64K area starting at this memory segment as long as you don't add a driver or memory-resident program. With this information you can determine where in memory to put a subroutine, or know exactly where yours loads if it's in Basic's data segment. You can then load Debug via the Shell statement and explore your subroutine *in situ*.

I have yet to figure out how to load Basic from Debug and run it with a stop point set at a subroutine, as I can on an IBM. It just doesn't stop. Let me know if you've found a way.

Sorry DeskMate

DeskMate doesn't work as an all-purpose text editor (I'm embarrassed to discover now). DeskMate can't write batch files or source files for Microsoft's assemblers and compilers.

GW-Basic is very forgiving, however, and takes listings DeskMate writes. DeskMate requires that you end text file names with the DOC extension or it won't load them.

I'll summarize. DeskMate text files are pure ASCII files with code 26 (1A hex) ending files. In true Tandy style, however, DeskMate's text editor uses only carriage return (ASCII 13) to end lines, and not CR/LF (13/10) as do MS-DOS programs. Edlin, MASM, and the DOS batch file processor expect 10 to follow every 13, but they'll take any character—I mean any—in its place. If you write a batch file with DeskMate and start each line after the first with a space, it'll run because they assume the extra character is 10. When DeskMate loads a text file written by Edlin (or others), it replaces the line feed code (10) with an ASCII space. Leave it there so Edlin will think it's a line feed character.

GW-Basic loads a program whether or not it finds the line feed code as is or replaced with a space. Don't try to load files created with the Copy command (COPY CON file name) with DeskMate because they don't end with an ASCII 26 (code for end-of-file). So you can use DeskMate if you're in a bind. ■

Address your correspondence to Dave Rowell, 80 Micro, 80 Pine St., Peterborough, NH 03458.

Lots of Ware

For $79.95 you can order Tryware Volume 1, 16 programs on 10 disks for the IBM PC/XT and compatibles (including the Models 1000 and 1200).

Volume 1 includes eight applications (a word processor, spreadsheet, data base manager, communications software, file cataloging program, menu manager, file security program, and keyboard enhancer), three utilities (disk file modification program, file directory manager, and a WordStar conversion program), and five games.

Most of the programs are user supported, which means that you're expected to register for extra benefits (like free updates, expanded manual and telephone support) by sending an additional contribution of $5–$75 to the author if you like the program.

Each program includes documentation. For more information, contact Briter Inc., 1100 E. Hector St., Conshohocken, PA 19428, 215-828-3230.

Circle 576 on Reader Service card.

Ten disks and 16 programs from Briter Inc.

Perfect Drawing

Microdex Corporation offers new versions of its xT.CAD ($449.95) computer-assisted drafting software for the Models 1000 and 1200. It's a general-purpose scaled technical drafting system for engineers, architects, and manufacturers, as well as an educational resource for schools and colleges.

Single-stroke mnemonic keyboard input creates drawings immediately on the screen. XT.CAD includes overlay, zoom, pan, block copy/rotate, text label commands, and more.

The program also offers several performance improvements, including expanded scaling systems, easier numerical input options, and additional messages. You need 256K, two disks, a graphics adapter (included on Model 1000), and an RS-232 interface for a plotter.

Versions of xT.CAD are also available for the Models III and 4/4P. For more information, contact Microdex Corp., 1212 N. Sawtelle, Tucson, AZ 85716, 602-326-3502.

Circle 575 on Reader Service card.

Memory Plus

A multifunction board from Matthew Electronics inc. gives your Models 1000 and 1200 access to a megabyte of memory, addressed as two banks of 512K bytes each.

The board is available in two models. The $545 MEI-1000P includes a megabyte of RAM plus a selectable direct-memory access (DMA) controller, a clock, calendar, and port expansion interface. You can turn the DMA controller on or off with a switch on the board. The MEI-1000S ($555) has a serial interface you can configure for an RS-232C port or an RS-422 port.

You can get both boards with 512K of RAM ($385 and $395) and later upgrade them to a full megabyte.

For more information, contact Matthew Electronics inc., 386 Avenida de la Vereda, Ojai, CA 93023, 805-646-7790.

Circle 573 on Reader Service card.

Better Accounts

Dac Software Inc. has updated its Dac-Easy Accounting software to accommodate both inventory and service-oriented businesses. The Model 1000/1200/2000 package includes general ledger, accounts receivable, accounts payable, purchase order, inventory, billing, and forecasting programs.

Dac-Easy's updated version also includes customized reports for purchase orders, invoices, and statements, and expanded codes for non-inventory items. The new price is $69.95. For more information contact Dac Software Inc., 4801 Spring Valley Road, Building 110-B, Dallas, TX 75244, 214-458-0038.

Circle 570 on Reader Service card.

Back It Up

Fullback ($88 + $3 shipping) from Alps, a hard or floppy disk back-up system for the Models 1000, 1200, and 2000, automatically backs up files in one, several, or all subdirectories with a single command. Options on the command line let you back up modified files only, all files, and files by date or alphabetical range.

If a directory is too large for a floppy disk, Fullback lets you selectively back up file groups across multiple floppies. If a file is too large for a floppy disk, you can back it up across multiple disks and later restore its original order. For more information, contact Alps, 1502 County Road 25, Woodland Park, CO 80863, 303-687-1442.

Circle 572 on Reader Service card.

Interface Mania

CMB3 Technologies' program for MS-DOS computers, The President, lets you access all capabilities of any printer through any application program or utility.

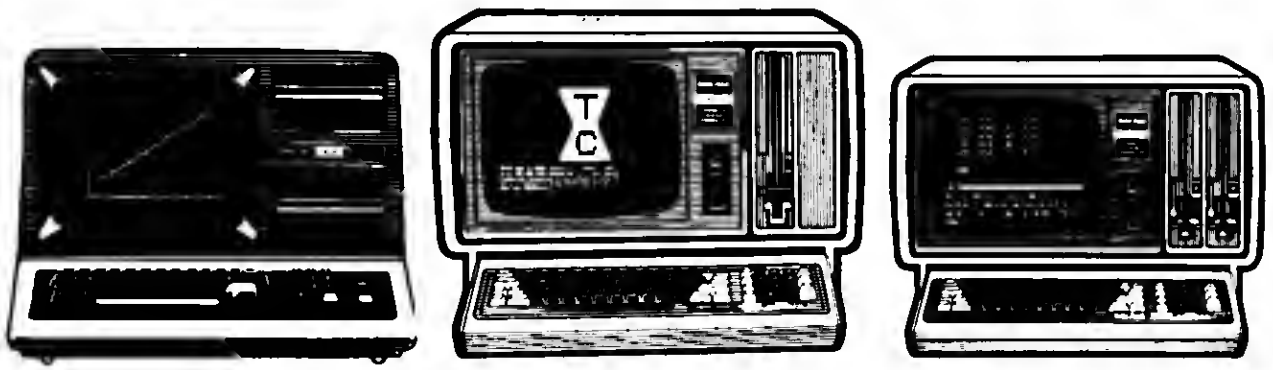
The memory-resident program lets you select character style, pitch, or effect (like superscripts, subscripts, underlines, and so on) with any word processor. You can print wide, bold titles; change character width and style; and underline totals in any spreadsheet.

The President ($99 plus shipping) comes with full documentation, demonstrations, and examples. For fur-

NOCONA

ELECTRONICS

MODEL 4 **MODEL 6000** **MODEL 12**



256K MOD 2000 W/MONO MONITOR	1319.00	CGP 220 PRINTER	499.00
256K MOD 2000 W/COLOR MONITOR	1629.00	DMP 105 PRINTER	145.00
256K MOD 1200 HD 1 DR.	1499.00	DMP 130 PRINTER	249.00
128K MOD 1000 W/MONO 1 DR.	779.00	DMP 430 PRINTER	585.00
128K MOD 1000 W/COLOR 2 DR.	1119.00	DMP 2100P PRINTER	1019.00
256K MOD 1000 W/10 MEG HD	1439.00	DWP 220 PRINTER	429.00
24K MOD 100 PORT	469.00	DWP 510 PRINTER	999.00
24K MOD 200 PORT	679.00	TRACTOR DWP 220	99.00
64K MOD 4D 2DR.	839.00	TRACTOR DWP 11/510	159.00
512K MOD 6000 2DR.	3099.00	TRACTOR FEED DMP2100P	99.00
512K MOD 6000 W/15 MEG HD	3799.00	PARALLEL PRINTER SWITCH	96.00
15 MEG HARD DISK	1329.00	PARALLEL PRINTER CONTROLLER	179.00
35 MEG HARD DISK	2059.00	70 MEG HARD DISK	2949.00

NEW DMP 2200 PRINTER 1149.00

100% RS COMPONENTS NO FOREIGN DRIVES OR MEMORY FULL WARRANTY
ALL RS SOFTWARE 20% OFF CATALOG PRICE
CASHIERS CHECK OR MONEY ORDER MUST ACCOMPANY ALL ORDERS

(817) 825-4027

NOCONA ELECTRONICS • BOX 593 • NOCONA, TX 76255

MS-DOS NEW PRODUCTS

Run your business from a hard disk with Hardisk Accounting Series.

ther details, contact CM83 Technologies, P.O. Box 3061, Walnut Creek, CA 94598, 415-930-0470.
Circle 574 on Reader Service card.

Hard Disk Accounting

Great Plains Sofiware develops, manufactures, and sells the Hardisk Accounting Series for the Models 1000, 1200, and 2000.

The Accounting Series comprises general ledger, accounts receivable, accounts payable, payroll, inventory management with point of sale invoicing, and job cost programs. You can start with one program like General Ledger, then add others like Accounts Payable or Inventory as you need them.

The Hardisk Accounting Series is available in both single- ($695 per module) and multi-user editions ($795 per module). You need 256K memory and a 5-megabyte hard disk. Contact Great Plains Software, 1701 S.W. 38th St., Fargo, ND 58103, 701-281-0550.
Circle 577 on Reader Service card.

Book Works

Interactive Bibliography ($99) for MS-DOS machines from SourceView Software International is a specialized data base management system for bibliographic reference files.

When you enter a reference in the bibliography data base, you identify up to 10 key words by which you can sort. In addition, you can sort an entire file by any field.

For details, contact SourceView Software International, 835 Castro St., Martinez, CA 94553. 800-443-0100.
Circle 571 on Reader Service card.

MS-DOS New Products Index

Reader Service Number	Company	Page
572	Alps	102
576	Briter Inc.	102
574	CMB3 Technologies	102
570	Dac Software Inc.	102
577	Great Plains Software	103
573	Matthew Electronics Inc.	102
575	Microdex Corp.	102
571	SourceView Software International	103

New Products listings are based on information supplied in manufacturers' press releases. 80 Micro has not tested or reviewed these products and cannot guorontee ony claims.

Circle 151 on Reader Service card

CONVERT YOUR TRS-80 MODEL III OR 4 INTO A DEVELOPMENT SYSTEM

Now you can develop Z-80 based *stand alone* devices such as games, robots, instruments and peripheral controllers, by using your TRS-80 as a development system. The DEVELOPMATE plugs into the expansion connector of your TRS-80 and adds **PROM PROGRAMMING** and **IN-CIRCUIT-EMULATION** capabilities to your system.

Complete instructions and sample schematics are included to help you design your own simple stand-alone microcomputer systems. THESE SYSTEMS CAN BE AS SIMPLE AS FOUR ICs: one TTL circuit for clock and reset, a Z-80, an EPROM, and one peripheral interface chip.

When the In-Circuit-Emulation cable is plugged into the Z-80 socket of your stand alone system, the system becomes a part of your TRS-80. You can use the full power of your editor/assembler's debug and trace programs to check out both the *hardware and the software*. Simple test loops can be used to check out the hardware, then the system program can be run to debug the logic of your stand-alone device.

Since the program is kept in TRS-80 RAM, changes can be made quickly and easily. When your stand-alone device works as desired, you use the Developmate's PROM PROGRAMMER to copy the program into a PROM. With this PROM and a Z-80 in place of the emulation cable, your stand alone device will work by itself!

The DEVELOPMATE is extremely compact. Both the PROM programmer and the In-Circuit-Emulator are in one small plastic box only 3.2" x 5.4". A line-plug mounted power supply is included. The PROM programmer has a "personality module" which defines the voltages and connections of the PROM so that future devices can be accommodated. However, the system comes with a "universal" personality module which handles 2758, 2508(8K); 2716,2516(16K); 2532(32K); as well as the new electrically alterable 2816 and 48016(16K EEPROMs).

The COMPLETE DEVELOPMATE 83 with software, power supply, emulation cable, TRS-80 cable, and "universal" personality module, is ONLY $329!

PM2 PERSONALITY MODULE for 2732A EPROM $15
PM3 PERSONALITY MODULE for 2764 EPROM $15

ORION INSTRUMENTS
702 Marshall Street #614
Redwood City, CA 94064
415-361-8883
MasterCard and Visa phone orders accepted.
California residents please add sales tax.

Circle 250 on Reader Service card.

MARYMAC INDUSTRIES INC.

800-231-3680

Radio Shack® Tandy®

EPSON PRINTERS

People you Trust to give you the very best!

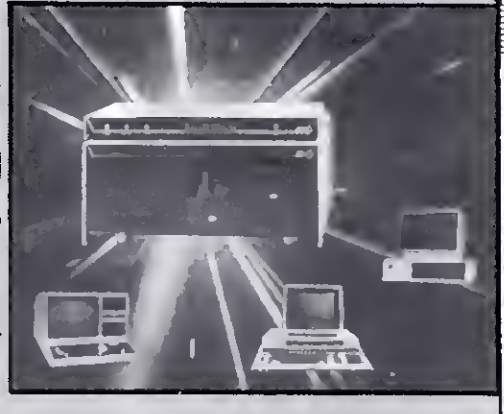
- Lowest Discount Prices
- Reliable Service
- Quality Products

"Worlds Largest Independent Authorized Computer Dealer"

22511 Katy Fwy., Katy (Houston) Texas 77450
(713) 392-0747 Telex 774132

Circle 300 on Reader Service card.

The Bi-Tech Star Network



Able to link 8 computers to a Hard Drive System

For: IBM PC, XT, AT, Compaq, AT&T PC 6300, PC compatibles, & also TANDY, EPSON & NEC computers.

RADIO SHACK MODEL FEATURES:
- 10 Megabyte Removable Cartridge
- 10-60 Megabytes Drive
- Multiple Operating System Segments
- Variable Segment Size Capability

MODEL I/III/4 SYSTEMS –
- DosPlus 3.5 & DosPlus 4
- LDos 5.1
- TRSDOS 6
- CP/M – Hard Disk Support

MODEL II/12 SYSTEMS –
- DP-II
- TRSDOS 2.0 with Racet HSDS
- CPM – Hard Disk Support

MODEL 16 –
- Xenix Support

MODEL 1000/1200/2000 –
- MS DOS, PC DOS

SPECIAL CLOSEOUT

2 user MULTIPLEXER (EXPANDABLE TO 4 USERS)
for Models I/III/4 includes 2 Host Adaptors
FOR Bi-Tech Hard Drives
TANDY Hard Drives,
most manufacturers
$495.00

SPECIAL with any purchase of Bi-Tech MULTIPLEXER SYSTEM.
10 Megabyte Hard Disk at **$995.00** complete

Tandy 1000
Internal Hard Disk Upgrade Kits Includes: Controller, Drive, Cables & Manual

10 meg $595.00
20 meg $795.00

BT ENTERPRISES

B.T. Enterprises
10 Carlough Road
Bohemia, NY 11716-2996
(516) 567-8155 (800) 645-1165
B.T. Enterprises is a division of Bi-Tech Enterprises Inc.

All prices above include 2% cash discount
Credit card customers add 2% to prices
Add $4.00 shipping & handling on all prepaid orders
Add $7.00 shipping & handling on C.O.D. orders
MasterCard Visa

REFER TO DEPT. 1L

They Went Thataway: Controlling Program Flow With If. . .Then Tests

"If the payroll wagon arrives by noon, we'll stick up the mine office at 12:30, when the guards go eat," a burly bandit told his gang of B-western heavies on my TV the other night.

"But boss, what if the wagon's late?" asked one of the less dim-witted ones.

"Then we'll get some vittles, too, and pull the holdup at 1:30. If the wagon ain't here by then, we'll wait in the shade."

Mutters of approval. Break for a commercial.

I was charmed by this exchange, for it was a perfect example of an If. . .Then proposition in Basic. You could write a simple Basic listing to simulate the events of the gang's plan.

The Decision-Maker

In Basic, you use If. . .Then tests to trigger new events if current events fulfill stated conditions. This test opens nearly infinite possibilities: If a specified Basic event occurs, then you can do anything else of which Basic is capable. And I mean anything!

Let's start small:

```
100 CLEAR: CLS
110 FOR X = 1 to 10
120 PRINT X
130 IF X = 5 THEN END
140 NEXT X
150 END
```

The key to this program lies in line 130. If X has attained a value of 5, then the program ends. You could change line 130 to anything else in Basic:

```
IF X = 1 THEN A$ = "IT'S 1 P.M.; IF X = 3 THEN Y = 2;
IF X > 1 THEN GOSUB 1000
IF X < > 5 THEN PRINT "X IS NOT EQUAL TO 5"
IF X = Y THEN A$ = A$ + STR$(Y)
```

The If statement tests any Basic event, and the Then statement fosters any Basic event:

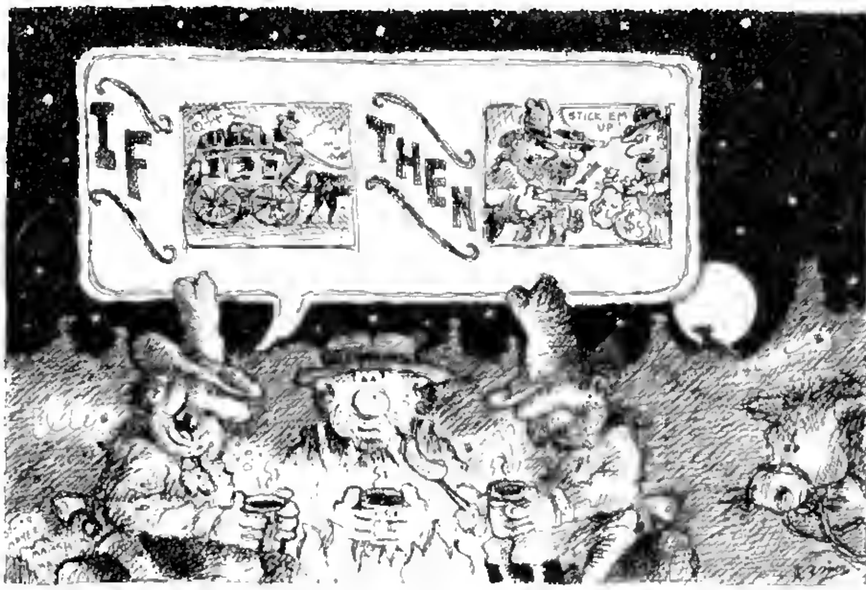


Illustration by Vinoy Laughner

```
IF A$ = "APPLESAUCE" THEN PRINT "I WANTED ICE CREAM."
IF Z$ = "Y" THEN MERGE "CUSTER/BAS"
IF L = 1 THEN PRINT "I'M SORRY. THAT IS WRONG."
IF INKEY$ < > "" THEN PRINT "HEY, I TOLD YOU NOT TO TOUCH THAT KEYBOARD!"
```

You should realize that when program values fulfill an If test, everything requested past the Then will happen. Should conditions not meet the test, Basic ignores the Then events. This exemplifies a common If. . .Then programming mistake: making essential program code dependent on the If test. Here's an example:

```
100 CLEAR: CLS
110 FOR X = 1 TO 5
120 PRINT X
130 IF X >3 THEN PRINT X"IS MORE THAN 3": NEXT X
140 END
```

The intent of this program is to go through a For. . .Next loop from 1 to 5, each time printing the value of X and noting when the value exceeds 3. It won't work because the NEXT X happens only if X is more than 3, and the incorrect If. . .Then test thwarts that possibility. To fix it, remove the NEXT X from the end of line 130 and give it its own line: 135 NEXT X.

Any time you get crazy results with an If. . .Then test, check whether you've included some event fundamental to program flow in the realm of Then events. And remember that everything past the Then statement occurs only when program conditions meet the If test.

Multiple Events

So far, I've covered one-element If tests. An If test can also stipulate multiple events, all of which must be met for the program to execute the event:

```
IF X = 1 AND Y = 2 AND B$ = "ZINGER" THEN PRINT "YOU WIN.": END
```

An If test can trigger a Then result if a program meets either of two or any of many tests:

```
IF X = 1 OR Y < = 30 OR G$ = "GOLLY" THEN PRINT "TEST MET."
```

You can combine these two forms:

```
IF X = 1 AND Y = 2 OR Z = 3 THEN PRINT "OK"
```

In this example, Basic prints "OK" if Z equals 3. It also prints "OK" if X equals 1 and Y equals 2. Consider another form of the If. . .Then test:

```
IF X = 1 OR Y = 2 AND Z = 3 THEN PRINT "OK"
```

In this example, Basic prints "OK" if X equals 1 or if Y equals 2 and Z equals 3. To understand these concepts better, think of the Or statement as a wall between possibilities, and the And statement as a bridge.

System Requirements

Models I, III, 4, 100, 1000, 1200, and 2000
Basic

Program Listing 1. Factors.

```
100 REM * FACTORS
110 CLEAR: CLS
120 FOR X=1 TO 25
130 PRINT "Factors of"X
140 FOR Y=1 TO X
150 IF X/Y=INT(X/Y) THEN PRINT Y;
160 NEXT Y: PRINT
170 PRINT "TAP A KEY TO CONTINUE"
180 X$=INKEY$
190 IF X$="" THEN 180
200 CLS: NEXT X: END
220 CLEAR: CLS
```

End

Program Listing 2. Heads-Tails.

```
100 REM * Heads-Tails
110 CLEAR: CLS
120 A=RND(2)
130 IF A=1 THEN H=H+1: GOTO 120
140 IF H>H THEN S=H: PRINT S
150 H=0: GOTO 120
160 END
```

End

You can put a series of If. . .Then tests on one program line. Any time the programs fails to meet a test, it falls through to the next line for further instructions. As long as a program meets conditions of the tests, it gives Then results and makes subsequent If tests.

Here's an example:

```
IF X = 1 THEN PRINT "YES": IF Y = 1 THEN PRINT "SI": IF Z = 1 THEN PRINT "JA"
```

The program won't test for Z unless X and Y both equal 1. It won't test for Y unless X is 1. And nothing happens if X doesn't equal 1. You'll find cases in which it's useful to isolate fall-through tests such as these on the same line.

Putting If. . .Then to Work

One of the best uses of an If. . .Then test is in working with factors, numbers evenly divisible into larger ones. Program Listing 1, Factors, uses If. . .Then to test and print factors for the numbers 1–25. The crucial test occurs in line 150. In another If. . .Then test, line 190 keeps the current results on-screen until you tap any key to continue.

Factors represents an example of letting a computer do the drudge work while you relax. You could amend it to print out the factors for the numbers from 1 to as high as the computer accepts. And I hope it suggests some possibilities for problem-solving and answer-finding using programs that automatically seek, sift, save, compare, contrast, and so on.

Program Listing 2, Heads-Tails, uses two If. . .Then tests. Line 120 simulates the flip of a coin. In line 130, if A equals 1, the program accepts it as heads and increments the heads total (variable H) by 1. I wrote this line to accept only consecutive occurrences of heads. If you get a tail, line 140 tests whether you set a record for a consecutive run of heads and, if so, assigns a new high score to variable S. When you run this program, it's unlikely you'll get more than seven or eight straight occurrences of heads, unless you let the program run a long time.

Program Listing 3. Alphabytes.

```
100 REM * Alphabytes *
110 CLEAR: CLS
120 FOR X=1 TO 2
130 INPUT "Type a word and press Entar";A$(X)
140 NEXT X
150 IF A$(1)<A$(2) THEN PRINT A$(1); ELSE PRINT A$(2);
170 PRINT " is alphabetically first"
180 END
```

End

A Matter of Relations

I was amazed when I realized that programmers write most If. . .Then tests with just a few relational operators. They are equal to (=), less than (<), and greater than (>). In combining these we come up with not equal to (< > or > <), less than or equal to (= < or < =), and more than or equal to (> = or = >).

You can use these symbols to test numbers and strings. You probably have a good command of number tests, but consider how you can use string tests. An alphabetical sort program works by comparing the ASCII values of character strings. Try Program Listing 3, Alphabytes. (A true sort program is more complex than Alphabytes; it passes through a list of words many times, swapping values until the list is in order.)

An If. . .Then test that also includes the Basic command Else gives you a way for either of two Then events to occur—one when the program meets the if test, the other when it doesn't:

```
IF X = 1 THEN PRINT "YES" ELSE PRINT "NO"
```

You can also obtain multiple Else results:

```
IF X = 1 THEN PRINT "ONE" ELSE IF X = 2 THEN PRINT "TWO" ELSE IF X = 3 THEN PRINT "THREE"
```

■

Write to Richard Ramella at 1493 Mt. View Ave., Chico, CA 95926.

Putting the Squeeze On Model 4 Programs

Many books and articles have spread the common misconception that Basic is a purely interpretive language. According to this point of view, Basic interprets program code as it executes each line.

Basic is indeed interpretive, but what it interprets while it executes a program or a direct command looks nothing like the code you write. As you type in each line of a program, Basic transforms it and, in a sense, precompiles it so that the computer can execute the line more quickly.

If you use Model I (Level II) or Model III Basic, either cassette- or disk-based, the computer translates the key words in each line into one of 128 possible tokens as soon as you press the enter key. This tokenizing scheme lets Basic execute a program relatively quickly because it already knows the commands in each line and doesn't have to look each one up in a table. Model I/III Basic represents each token within Basic as a single value between 80 and 0FF hexadecimal (hex).

Model 4 Basic uses more than 128 key words, and therefore needs to extend this translation scheme somewhat. Pages A-82 and A-83 in the TRSDOS 6.2 manual show all the key words in Model 4 Basic and the tokens, or internal codes, for each. Basic internally represents those tokens with values above 65000 as a byte of 0FF hex, which signals an extended-code key word, plus another byte specific to that key word or command.

You never notice the time Basic takes to translate the line you type into an internal, tokenized line of code because you type slowly by a computer's standards. When you do notice a pause after entering or editing a line, it's not because of interpretation but because Basic moves around program lines in memory. However, if Basic had to decipher each command in each program line during execution, your programs would run much more slowly than they do.

LOAD 80

System Requirements

Model 4
Disk Basic 01.01.00
Assembly language
Editor/assembler

Basic Differences

Model I/III Basic translates each key word into a token and leaves the rest of the program line unchanged. Basic's execution, or run-time, module must then do the rest of the necessary interpretations every time it executes each line.

Model 4 Basic operates differently. Not only does it translate all key words into tokens when you enter a line, it also translates all numeric values into Basic's internal format. Model I/III Basic recognizes four types of numbers: line numbers (zero to 65529), integers (–32768 to 32767), and single- and double-precision floating-point numbers. However, Basic holds all numbers in their literal, ASCII format inside program lines and translates them into an internal form during execution.

Model 4 Basic changes all numbers to an internal format at the same time it tokenizes each line; that is, when you enter the line. It recognizes nine types of numbers and uses a separate internal form for each. Except for one-character values (zero to nine) and numbers in data statements, Model 4 Basic adds a prefix to each numeric value to show what type of number it is (see the Table). Model 4 programs seem to list more slowly than their Model I/III counterparts partly because Basic must translate all numbers from their internal representations back to their external ASCII form.

Because Model 4 Basic translates numbers into and out of internal format, a line sometimes appears to have changed after you enter it. For example, the line:

```
10 A = &H000F: B = 12.0
```

will list as:

```
10 A = &HF : B = 12!
```

Basic hasn't changed either value, but the ASCII representation of each is different. The exclamation point at the end of the line shows that Basic interprets that 12 as a single-precision floating-point number rather than as an integer.

Model I/III Basic recognizes only the first two characters of a variable name as significant. Model 4 Basic recognizes the first 40 characters of the name as significant so that, for example, it sees PRICE and PROFIT as different variables. It also lets you include key words in variable names. A variable named FORM would be impossible in Model I/III Basic since it contains the key words FOR and OR; it is perfectly acceptable in Model 4 Basic.

To distinguish between *true* key words and key words accidentally included in

Illustration by Peter Bono

Circle 9 on Reader Service card.

H&E COMPUTRONICS INC.

FOR YOUR TRS-80™ • APPLE™ • IBM PC™ • ATARI™ • OSBORNE™ • XEROX™ • KAYPRO™ • ALL COMMODORE Computers TELEVIDEO™ • ZENITH™ • SANYO™ • NEC™ • DEC™ • TI PROFESSIONAL COMPUTER™ • SUPERBRAIN JR.™ • EPSON™ Any CP/M™ Computer with 8" Disk Drives

CP/M-based Computers must be equipped with Microsoft BASIC (MBASIC or BASIC-80)

TRS-80 trademark Tandy Corp · APPLE trademark Apple Corp. · IBM PC trademark IBM Corp. · ATARI trademark Atari, Inc. · OSBORNE trademark Osborne Corp XEROX trademark Xerox Corp · KAYPRO trademark Non-Linear Systems, Inc. · TELEVIDEO trademark Televideo Systems, Inc. · SANYO trademark Sanyo Corp. NEC trademark NEC Corp · DEC trademark Digital Equipment Corp. · ZENITH trademark Zenith Corp. · TI PROFESSIONAL COMPUTER trademark Texas Instruments, Inc SUPERBRAIN trademark Intertec Corp. · CP/M trademark Digital Research · EPSON trademark Epson Corp

NEW

BUSINESS PAC 100

100 Ready-To-Run Business Programs

(ON CASSETTE OR DISKETTE).....Includes 128 Page Users Manual..... Inventory Control.....Payroll.....Bookkeeping System.....Stock Calculations..... Checkbook Maintenance.....Accounts Receivable.....Accounts Payable.....

BUSINESS PAC 100 PROGRAM LIST

	NAME	DESCRIPTION
1	RULE78	Interest Apportionment by Rule of the 78's
2	ANNUI	Annuity computation program
3	DATE	Time between dates
4	DAYYEAR	Day of year a particular date falls on
5	LEASEINT	Interest rate on lease
6	BREAKEVN	Breakeven analysis
7	DEPRSL	Straightline depreciation
8	DEPRSY	Sum of the digits depreciation
9	DEPRDB	Declining balance depreciation
10	DEPRDDB	Double declining balance depreciation
11	TAXDEP	Cash flow vs. depreciation tables
12	CHECK2	Prints RAPIDFORMS checks along with daily register
13	CHECKBK1	Checkbook maintenance program
14	MORTGAGE/A	Mortgage amortization table
15	MULTMON	Computes time needed for money to double, triple, etc
16	SALVAGE	Determines salvage value of an investment
17	RRVARIN	Rate of return on investment with variable inflows
18	RRCONST	Rate of return on investment with constant inflows
19	EFFECT	Effective interest rate of a loan
20	FVAL	Future value of an investment (compound interest)
21	PVAL	Present value of a future amount
22	LOANPAY	Amount of payment on a loan
23	REGWITH	Equal withdrawals from investment to leave 0 over
24	SIMPDISK	Simple discount analysis
25	DATEVAL	Equivalent & nonequivalent dated values for oblig
26	ANNUDEF	Present value of deferred annuities
27	MARKUP	% Markup analysis for items
28	SINKFUND	Sinking fund amortization program
29	BONDVAL	Value of a bond
30	DEPLETE	Depletion analysis
31	BLACKSH	Black Scholes options analysis
32	STOCVAL1	Expected return on stock via discounts dividends
33	WARVAL	Value of a warrant
34	BONDVAL2	Value of a bond
35	EPSEST	Estimate of future earnings per share for company
36	BETAALPH	Computes alpha and beta variables for stock
37	SHARPE1	Portfolio selection model-i.e. what stocks to hold
38	OPTWRITE	Option writing computations
39	RTVAL	Value of a right
40	EXPVAL	Expected value analysis
41	BAYES	Bayesian decisions
42	VALPRINF	Value of perfect information
43	VALADINF	Value of additional information
44	UTILITY	Derives utility function
45	SIMPLEX	Linear programming solution by simplex method
46	TRANS	Transportation method for linear programming
47	EOQ	Economic order quantity inventory model
48	QUEUE1	Single server queueing (waiting line) model
49	CVP	Cost-volume-profit analysis
50	CONDPROF	Conditional profit tables
51	OPTLOSS	Opportunity loss tables
52	FQUOQ	Fixed quantity economic order quantity model
53	FQEOWSH	As above but with shortages permitted
54	FQEOQPB	As above but with quantity price breaks
55	QUEUECB	Cost-benefit waiting line analysis
56	NCFANAL	Net cash-flow analysis for simple investment
57	PROFIND	Profitability index of a project
58	CAPI	Cap. Asset Pr. Model analysis of project
59	WACC	Weighted average cost of capital
60	COMPBAL	True rate on loan with compensating bal. required
61	DISCBAL	True rate on discounted loan
62	MERGANAL	Merger analysis computations
63	FINRAT	Financial ratios for a firm
64	NPV	Net present value of project
65	PRINDLAS	Laspeyres price index
66	PRINDPA	Paasche price index
67	SEASIND	Constructs seasonal quantity indices for company
68	TIMETR	Time series analysis linear trend
69	TIMEMOV	Time series analysis moving average trend
70	FUPRINF	Future price estimation with inflation
71	MAILPAC	Mailing list system
72	LETWRT	Letter writing system-links with MAILPAC
73	SORT3	Sorts list of names
74	LABEL1	Shipping label maker
75	LABEL2	Name label maker
76	BUSBUD	DOME business bookkeeping system
77	TIMECLCK	Computes weeks total hours from timeclock info
78	ACCTPAY	In memory accounts payable system-storage permitted
79	INVOICE	Generate invoice on screen and print on printer
80	INVENT2	In memory inventory control system
81	TELDIR	Computerized telephone directory
82	TIMUSAN	Time use analysis
83	ASSIGN	Use of assignment algorithm for optimal job assign.
84	ACCTREC	In memory accounts receivable system-storage ok
85	TERMSPAY	Compares 3 methods of repayment of loans
86	PAYNET	Computes gross pay required for given net
87	SELLPR	Computes selling price for given after tax amount
88	ARBCOMP	Arbitrage computations
89	DEPRSF	Sinking fund depreciation
90	UPSZONE	Finds UPS zones from zip code
91	ENVELOPE	Types envelope including return address
92	AUTOEXP	Automobile expense analysis
93	INSFILE	Insurance policy file
94	PAYROLL2	In memory payroll system
95	DILANAL	Dilution analysis
96	LOANAFFD	Loan amount a borrower can afford
97	RENTPRCH	Purchase price for rental property
98	SALELEAS	Sale-leaseback analysis
99	RRCONVBD	Investor's rate of return on convertable bond
100	PORTVAL9	Stock market portfolio storage-valuation program

☐ TRS-80 Cassette Version ..... **$ 99.95**
☐ TRS-80 (Mod I/III/4/Color) Commodore, Apple or Atari Versions **$ 99.95**
☐ IBM, TRS-80 (Mod II/12/16), Kaypro, and CP/M Versions **$149.95**

30-DAY MONEY BACK GUARANTEE

H&E COMPUTRONICS INC.
50 N. PASCACK RD., SPRING VALLEY, N.Y. 10977

TOLL FREE ORDER LINE: (OUTSIDE OF N.Y. STATE) **(800) 431-2818**

24 HOUR ORDER LINE: **(914) 425-1535**

- ADD $3 for shipping in UPS Areas
- Add $4 for C.O.D. or Non-UPS Areas
- Add $5 to CANADA or MEXICO
- Add proper postage elsewhere

ALL PRICES AND SPECIFICATIONS SUBJECT TO CHANGE/DELIVERY SUBJECT TO AVAILABILITY

ASK FOR OUR 64-PAGE CATALOG

ALL ORDERS PROCESSED WITHIN 24 HOURS
DEALER INQUIRIES INVITED

Token	Meaning	Internal Form	ASCII Example
0A hex	(Line feed)		
0B hex	Octal number	0B nn nn	$01234
0C hex	Hex number	0C nn nn	&H1234
0D hex	(Carriage return)		
0E hex	Line number	0E nn nn	GOTO 1234
0F hex	Single byte (10–255)	0F nn	123
10 hex	(Apparently unused)		
11 hex	Single digit numbers	11	0
12 hex	.	12	1
.	.	.	.
.	.	.	.
19 hex	.	19	8
1A hex	.	1A	9
1B hex	(Apparently unused)		
1C hex	2-byte integer	1C nn nn	1234
1D hex	4-byte single-precision floating point	1D nn nn nn nn	1234.5
1E hex	(Apparently unused)		
1F hex	8-byte double-precision floating point	1F nn nn nn nn nn nn nn nn	1234.5#
20 hex	(ASCII space)		

Negative numbers use the same representation but are prefixed with the token for a minus sign, 0F4 hex.

Numbers in Data statements are stored unchanged in their original ASCII format.

Table. Model 4 Basic's internal representation of numeric values.

Program Listing. Squeeze filter.

```
00100 ;-----------------------------------------------------------------
00110 ;     Utility to SQUEEZE all unneeded spaces from
00120 ;       a BASIC program in memory.  Also removes
00130 ;       remarks and linefeed characters.  Does not
00140 ;       alter literal strings.
00150 ;
00160 ;     For BASIC 01.01.00 only!
00170 ;       Tested with TRSDOS 6.2 (see text)
00180 ;-----------------------------------------------------------------
00190 ;
00200 ;  SVCs used:
00210 @DSPLY          EQU     0AH
00220 @CHNIO          EQU     14H
00230 @EXIT           EQU     16H
00240 @GTMOD          EQU     53H
00250 @HEXDEC         EQU     61H
00260 @HIGH$          EQU     64H
00270 @FLAGS          EQU     65H
00280 ;
00290 ;  Other constants (see text):
00300 ;
00310 PRG_TBL         EQU     71A5H   ;==> Basic's program table
00320 VAR_TBL         EQU     719FH   ;==> variable table
00330 ARR_TBL         EQU     71A1H   ;==> array table
00340 FRE_SPC         EQU     71A3H   ;==> free space
00350 ;
00360 OCT             EQU     0BH     ;Octal number token
00370 HEX             EQU     0CH     ;Hex number token
00380 LINE            EQU     0EH     ;Line number token
00390 BYTE            EQU     0FH     ;Byte value token
00400 INTEGER         EQU     1CH     ;Integer value token
00410 SINGLE          EQU     1DH     ;Single-prec. value token
00420 DOUBLE          EQU     1FH     ;Double-prec. value token
00430 ;
00440 LF              EQU     0AH     ;Linefeed character
00450 CR              EQU     0DH     ;Carriage return character
00460 SPACE           EQU     20H     ;Space character
00470 REMARK          EQU     0FBH    ;REM token
00480 EXTEND          EQU     0FFH    ;Extended command token
00490 ;
00500 SIGNAL          EQU     0F3H    ;<clear><Shift><S>
00510 ;
00520 ;  Macro instructions
00530 ;
00540 SVC     MACRO   #NUM            ;This is pre-defined in ALDS
00550         LD      A,#NUM
00560         RST     28H
00570         ENDM
```

Listing continued

variable names. Model 4 Basic requires that you separate each key word and variable with some type of delimiter. You can use any character not allowed in a variable name—including a space, a comma, a parenthesis, an equals sign, and the math and relational operators—as a delimiter. The result is that Model 4 Basic programs tend to contain many more spaces than Model III programs.

I almost always use extra spaces, tabs, line feeds, and comments when I write a program to make debugging easier. However, Basic stores each of these characters according to its internal representation, making programs longer than needed both in memory and on disk. Some long programs begin to run out of memory space in the Model 4. One solution for that is a utility that condenses a debugged program into the least possible space to allow as much room as possible in memory when you run the program. Squeeze is such a program; it removes all spaces, all line feeds, and the text of all remarks (see the Program Listing). It does not, however, affect literal strings in your program.

The Big Squeeze

To use Squeeze, you must first install it with the Set command at TRSDOS Ready and use the Filter command to link it to the keyboard driver. If you assemble the program as Squeeze/FLT, you would install it with the following two lines:

```
SET *SQ SQUEEZE
FILTER *KI *SQ
```

The program will report that it has successfully installed itself and then relocate itself to protected high memory. You invoke it by pressing clear/shift-S when you want to compress a Basic program in memory.

Squeeze displays each line number as it compresses your program. If these numbers are out of order, something has gone wrong and you should reload your Basic program from disk before trying again. If the numbers are in order, Squeeze has successfully compressed the program.

If you list a compressed program, you might be surprised to see that it apparently still contains some spaces. The internal representation of the program won't have any, but Basic's listing module will put spaces wherever necessary in the version it shows you. What you see is the minimum number of spaces you could use to enter the program.

Also, Squeeze removes the text of all remarks, but keeps the Remark statements in case you have a GOTO or GOSUB to a line that begins with a remark. Any remark that originally began with an apostrophe will be shown as REM.

Circle 383 on Reader Service card.

LETTER-WRITER™

"Integrated WORD PROCESSING Power"

NEW IV VERSION

THE "BEST" SOFTWARE IS GUARANTEED! TRY IT and LIKE IT, or GET a REFUND

The machine code disk L-W is "A" rated by Allenbach's "SOFTWARE REPORTS" for:

EASY USE, DOCUMENTATION, FEATURES

- FORM LETTERS and LABELS on any printer.
- ADD and SUB bookkeeping and tax columns.
- LEGAL PAPER LINE NUMBERING, Centering.
- Copy, move, delete, insert, merge, select.
- Screen display same as printout, unmodified M I users can see and print up/lower case.
- MASS MAILER, graphics, cancel changes.
- SPLIT SCREEN typing line, see old and new versions, word wrap, disk warm start.
- Over 50 "one key" INSERTION PHRASES.
- ONE MODE: Type / Edit / Delete without switching modes — NO TRAINING PERIOD.

WE PAY TAX and USA SHIPPING. Try your L-W for 3 months. Like it or return it for a refund, less our shipping costs of: $3.50.

- Basic 18K TAPE system: $ 27.99
- Extra Feature 32K DISK: $ 47.99

Versions: MI, MIII, MIV

Specials: The B U G S Game Tape $ 9.95, Disk $ 11.95

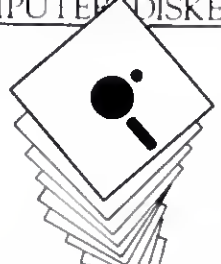
ASTRO-STAR ENTERPRISES
5905 Stone Hill Dr.
Rocklin, CA 95677

Information/orders:
(916) 624-3709

Orders: 1-800-622-4070
in IL 1-800-942-7317

Circle 295 on Reader Service card.

G T ENTERPRISES
COMPUTER DISKETTES



5¼
DS/DD
$.99 each

5¼
SS/DD
$.90 each

LIFETIME GUARANTEE

Individually Tested 100% Error Free
Reinforced Hub Rings Write Protect Tabs
Color ID Labels and Tyvec Envelopes Included

Add $3.00 shipping per 100 or fewer diskettes
WA Residents add 8% sales tax

PHONE ORDERS
C.O.D., MasterCard & Visa Accepted

MAIL ORDERS
Cashiers Checks & Money Orders accepted. Allow 2 weeks for delivery. Personal checks allow 3 weeks for clearance.

G T ENTERPRISES
2400-SW 325th Street, Federal Way, WA 98023
To Order, Phone Our 24 Hour Toll Free Line
1 (800) 828-2828, Ask For Ext. 525
Specific Questions and Inquiries
(206) 838-5107

Computer Stores ask about our wholesale and private labeling service

Circle 357 on Reader Service card.

MacInker™ Mercury™

Re-Ink Any Fabric Ribbon Automatically for Less Than 5¢

Now one **Universal Cartridge MacInker (UC)** re-inks all fabric cartridges and one **Universal Spool MacInker (US)** re-inks all spools. We support over 1000 printer brands.

Most dedicated **Mac Inker(s)** cost less than $60 and start at $54.95. The **Universal Cartridge MacInker** is $68.50. The **Universal Spool MacInker** is $66.95.

Use your **Mac Inker** to re-ink your dry cartridges (for less than 5 cents in ink) and watch the improvement in print-out quality. Our residueless, lubricated, dot matrix ink yields a darker print than most new ribbons. Or get any of our six basic ink colors: **Brown, Blue, Red, Green, Yellow, Purple** and use *Mac Inker* to *create* or *re-ink your own colored cartridges.* We have uninked or pre-colored cartridges.

Most cartridges can be used almost indefinitely, if ribbon is regularly re-inked, kept moist and lubricated, so that the fabric does not fray. Some customers have reported 80 re-inkings of the same cartridge and still getting excellent print.

As of October '85 we have more than 45,000 *Moc Inker(s)* in the field, in five continents (220V motors available). *Moc Inker* has been reviewed, approved and flattered in most magazines and even in the *New York Times* and the *Chicogo Sun Times.*

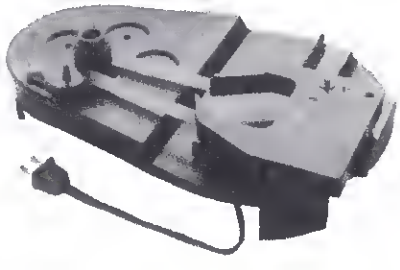
From the God of Communications a *Divine* Modem at a *Miracle* Price.

- **100% Hayes Smartmodem* Compatible;** that is, **Mercury** runs with all the popular and just about any private or public domain communications soltware.
- **300/1200 Baud Speed,** software or hardware (dip switch) selectable. The 1200 Baud features will save you a bundle in long-distance connections.
- **Auto Dial/Auto Answer/Speed Selection**
- **Audio Monitor** via speaker with volume control
- **Front Panel Lights** give you at a glance full information on the status and mode of your communication: modem ready — terminal ready — carrier detect — send data — receive data — high speed — auto answer — off hook.
- **Clear and Easy-to-Read Manual.** If modeming is new for you, the **Mercury** manual will ease your way into the exciting and rewarding world of data transmission.
- **18-Month Warranty**

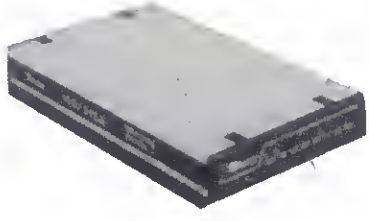
$265.00

Including power supply, telephone cable and manual. Computer to modem cable ($15) — shipping $4 anywhere in continental U.S.

*Hayes is a Trademark of Hayes Microproducts.



Universal Cartridge Mac Inker



Computer Friends

Order toll free 1-800-547-3303
or ask for free brochure

In Oregon and for 24-hour service (503) 297-2321.
Computer Friends
6415 SW Canyon Ct., #10
Portland, OR 97221

Dealer Inquiries Welcome

Listing continued

```
00580 ;
00590 STORE   MACRO                   ;Saves contents of DE in (HL)
00600         LD      (HL),E
00610         INC     HL
00620         LD      (HL),D
00630         ENDM
00640 ;
00650 ;-----------------------------------------------------------------
00660 ;  Memory-resident code
00670 ;-----------------------------------------------------------------
00680 ;
00690         ORG     3000H           ;Use PSECT 3000H with ALDS
00700 ;
00710 ;  Filter header:
00720 ;
00730 BEGIN   JR      START           ;Jump over header
00740 OLDHI   DEFW    $-$             ;2-bytes for old HIGH$
00750         DEFB    MODDCB-BEGIN-5  ;Length of module name
00760         DEFM    'SQUEEZE'       ;Module name
00770 MODDCB  DEFW    $-$             ;2-bytes for DCB address
00780         DEFW    0               ;Reserved by TRSDOS
00790 ;
00800 ;  Storage area
00810 ;
00820 NUMBUF  DEFS    5               ;5 spaces for ascii numbers
00830         DEFB    0DH             ;End with carriage return
00840 BASMSG  DEFM    'BASIC is not loaded!'
00850         DEFB    0DH
00860 ERROR   DEFM    'Program pointer error'
00870         DEFB    0DH
00880 ;
00890 ;  Basic Signature at 3000H:
00900 ;
00910 SIGNAT  DEFB    0E4H,0E2H,27H,0F1H,0ECH,00H,2EH,0F1H
00920                                 ;Use separate DEFB statements
00930                                 ;For assemblers other than EDAS
00940 ;
00950 ;  Link to *KI driver:
00960 ;
00970 CHAIN   PUSH    IX              ;Save old value
00980         LD      IX,(MODDCB)     ;Get our DCB address
00990 REL1    EQU     $-2
01000         SVC     @CHNIO          ;Move down chain
01010         POP     IX              ;Recover old value
01020         RET
01030 ;
01040 ;  Start of filter code:
01050 ;
01060 START   JR      NC,CHAIN        ;Go if not GET request
01070         CALL    CHAIN           ;Else go and return
01080 REL2    EQU     $-2
01090         RET     NZ              ;Return if no key
01100         PUSH    AF              ;Else save char & flags
01110         CP      SIGNAL          ;Our turn?
01120         JR      Z,GO            ;Yes -- start our routine
01130         POP     AF              ;Else recover flags
01140         RET                     ;And leave
01150 ;
01160 ;  Our routine has been invoked:
01170 ;
01180 GO      PUSH    BC              ;Save all registers
01190         PUSH    DE
01200         PUSH    HL
01210         PUSH    IX
01220         PUSH    IY
01230 ;
01240         LD      HL,3000H        ;HL==> beginning of program area
01250         LD      DE,SIGNAT       ;DL==> signature comparison table
01260 REL3    EQU     $-2
01270         LD      B,8             ;Bytes to check
01280 CKLOOP  LD      A,(DE)          ;Get signature byte
01290         CP      (HL)            ;Okay?
01300         JR      NZ,NOBAS        ;Go if not
01310         INC     HL              ;Else bump pointers
01320         INC     DE
01330         DJNZ    CKLOOP          ;Check 8 bytes
01340         JR      OKAY            ;And go
01350 ;
01360 ;  Basic is not resident
01370 ;
01380 NOBAS   LD      HL,BASMSG       ;Point to message
01390 REL4    EQU     $-2
01400         DEFB    0DDH            ;LD IX prefix
01410 INTERR  LD      HL,ERROR        ;Point to message
01420 REL5    EQU     $-2
01430         SVC     @DSPLY          ;Display it
01440         JP      OUT             ;And leave
01450 REL6    EQU     $-2
01460 ;
01470 ;  Basic is in memory -- start squeeze
01480 ;
01490 OKAY    LD      IX,(PRG_TBL)    ;IX==> User program
01500         LD      IY,(PRG_TBL)    ;IY==> User program
01510         CALL    BUMP1           ;Move byte from (IX) to (IY)
01520 REL7    EQU     $-2
01530         OR      A               ;Was it 00 line separator?
01540         JR      NZ,INTERR       ;No -- Report error & stop
01550 ;
```

Listing continued

which looks like a mistake but isn't. Basic normally stores an apostrophe used as a Remark command as 3 bytes: a colon to indicate a new command, a remark token, and a special token for the apostrophe itself. The compression utility removes the apostrophe token and all the text that follows the remark symbol, but leaves the colon and first remark token in place so that the program runs without error.

If you save a compressed program to disk in normal, tokenized form, it won't have any spaces. If you save it in ASCII form, it will contain the spaces you see when you list it. You can, of course, reload and run either form. If you edit a compressed line, the editor will put the spaces back in and you might want to compress the program again.

Before you assemble the Listing, you need to check the four values in lines 310–340. These are the addresses where Basic stores pointers to its program table (the list of program lines precompiled into internal format), its variable table, its array table, and the beginning of free space. To check those values, type in the following, beginning at TRSDOS Ready. End each line by pressing the enter key:

```
DEBUG (E)
BASIC.BASIC
G
10 '******
```

Now hit the break key, type in D8000, and press the enter key.

You have just entered a short Basic program that consists of a line number, a remark, and six asterisks, then entered Debug to see where the program resides. (You can return from Debug to Basic at any time by typing in G and pressing the enter key.)

You should see asterisks in the middle of the Debug display. If not, press the plus sign until the asterisks appear. When they do, look for the three 00 bytes that precede the asterisks and write down the memory address of the last of those three bytes.

Now type in D7100. Starting at 71A7 hex should be a series of 26 bytes of 04 hex. These indicate that all variables default to type 4, single-precision numeric. If you use a DEFINT, DEFSTR, or DEFDBL command in your program, some or all of those bytes will change.

The 2 bytes immediately before the series of 04s should contain the address you just wrote down, but in reverse order. For example, if you wrote down 8135, you should see 35 81. If that value resides at 71A5 and 71A6 hex, you can assemble the program without change. If it isn't, you'll have to search through memory (use the plus and minus keys) looking for the 26 04s immediately preceded by the address you wrote down. When you find

The squeeze filter checks the area starting at 3000 hex to see if you have Basic active.

it, you need to change the values in lines 310–340. Line 310 contains the address of the pointer to the beginning of your program. Lines 320, 330, and 340 are the addresses of three pointers that immediately precede that one. You probably won't have to change anything if you're using TRSDOS 6.2 and Basic 1.1.0.

To understand how Squeeze works, you need to know how Basic stores program lines internally. Each line begins with the 2-byte address of the next line, which gives the program the form of a forward-linked list. Following that are 2 bytes that contain the line number in normal LSB/MSB (least-significant byte/most-significant byte) form. The tokenized form of the contents of the line follow the line number. Basic separates each line from the next with a single 00 byte. The entire program ends with 3 successive bytes of 00: The first is the line separator; the next two (which would normally be the link field) show that the line links to no other line.

I've used two macro instructions in this program. The first makes using supervisory calls easier, the second stores the contents of the DE register pair at the address to which HL points. If your assembler doesn't support macros, you can easily expand each by hand. Next month, I plan to discuss macros in detail, including methods of expansion.

Program Operation

The code beginning in line 730 represents a standard TRSDOS memory header that allows TRSDOS to find modules in memory, and perform link, route, and filter operations successfully. Following that is a small buffer for converting line numbers to ASCII and two brief error messages.

Line 910 (you might have to write several separate DEFB statements with some assemblers) contains the first 8 bytes of the Basic/CMD program (you can verify they are correct with Debug) stored at 3000 hex. Most programs load into memory starting at 3000 hex and the Squeeze filter checks that area to see if you have Basic active. However, it is possible for those bytes to still reside in memory after you load and then leave

Listing continued

```
01560 LOOP1   PUSH    IY              ;Save address of memory link
01570         LD      B,2             ;Get 2 characters
01580         CALL    BUMP            ; from (IX) to (IY)
01590 REL8    EQU     $-2
01600         OR      A               ;Last one 00?
01610         JP      Z,OONE          ;Yes -- we're done
01620 REL9    EQU     $-2
01630 ;
01640         LD      L,(IX)          ;Get LSB of line number
01650         LD      H,(IX+1)        ;Get line number
01660         LD      DE,NUMBUF       ;DE==> buffer for ascii value
01670 REL10   EQU     $-2
01680         SVC     @HEXDEC         ;Convert to decimal
01690         LD      HL,NUMBUF       ;HL==> ascii string
01700 REL11   EQU     $-2
01710         SVC     @DSPLY          ;Display on screen
01720         LD      B,2             ;Move 2-byte line number
01730         CALL    BUMP            ;  from (IX) to (IX)
01740 REL12   EQU     $-2
01750 ;
01760 ;  Now scan line of Basic until line separator is found
01770 ;
01780 LOOP2   LD      A,(IX)          ;Get next byte
01790         OR      A               ;Is it 00 line separator?
01800         JR      Z,EOL           ;Yes -- go
01810         CP      REMARK          ;REM token?
01820         JR      Z,REM           ;Yes -- go
01830         CP      '"'             ;Beginning a string?
01840         JR      Z,STRING        ;Yes -- go
01850         CP      EXTEND          ;2-byte verb token?
01860         JR      NZ,GO1          ;No -- jump ahead
01870         LD      B,2             ;2 bytes to transfer
01880         CALL    BUMP            ;  from (IX) to (IY)
01890 REL13   EQU     $-2
01900         JR      LOOP2           ;And loop back
01910 GO1     CP      SPACE           ;A space?
01920         JR      NZ,GO3          ;No -- jump ahead
01930 GO2     INC     IX              ;Bump source pointer
01940         JR      LOOP2           ;And loop back
01950 GO3     JR      NC,XFER         ;Go if greater than a space
01960         CP      LF              ;Line feed character?
01970         JR      Z,GO2           ;Yes -- go.
01980         CP      OCT             ;Octal token?
01990         JR      Z,INT           ;Yes -- transfer 3 bytes
02000         CP      HEX             ;Hex token?
02010         JR      Z,INT           ;Yes -- transfer 3 bytes
02020         CP      LINE            ;Line number token?
02030         JR      Z,INT           ;Yes -- transfer 3 bytes
02040         CP      BYTE            ;Byte token?
02050         JR      Z,BYT           ;Yes -- go
02060         CP      INTEGER         ;Integer token
02070         JR      Z,INT           ;Yes -- go
02080         CP      SINGLE          ;Single-precision token
02090         JR      Z,SING          ;Yes -- go
02100         CP      DOUBLE          ;Double-precision token
02110         JR      NZ,XFER         ;No -- transfer one byte
02120         LD      B,9             ;Bytes to transfer
02130         JR      XFERB           ;And go
02140 SING    LD      B,5             ;Transfer single-precision
02150         JR      XFERB           ;And go
02160 INT     LD      B,3             ;Transfer integer
02170         JR      XFERB           ;And go
02180 BYT     LD      B,2             ;Transfer byte value
02190 XFERB   CALL    BUMP            ;Transfer number in B
02200 REL14   EQU     $-2
02210         JR      LOOP2           ;Scan some more
02220 ;
02230 XFER    CALL    BUMP1           ;Move one byte from (IX) to (IY)
02240 REL15   EQU     $-2
02250         JR      LOOP2           ;And scan some more
02260 ;
02270 ; Transfer a string
02280 ;
02290 STRING  CALL    BUMP1           ;Move opening quote
02300 REL16   EQU     $-2
02310 STR1    CALL    BUMP1           ;Move one character
02320 REL17   EQU     $-2
02330         OR      A               ;EOL Mark?
02340         JR      Z,EOL1          ;Yes -- go
02350         CP      '"'             ;Closing quote?
02360         JR      NZ,STR1         ;No -- loop back
02370         JR      LOOP2           ;And scan some more
02380 ;
02390 ; Transfer & Truncate a remark (leave REM token in place)
02400 ;
02410 REM     CALL    BUMP1           ;Move REM character
02420 REL18   EQU     $-2
02430 REM1    LD      A,(IX)          ;Get next character
02440         OR      A               ;Line separator?
02450         JR      Z,EOL           ;Go when end found
02460         INC     IX              ;Else bump pointer
02470         JR      REM1            ;And look some more
02480 ;
02490 ;  Process End-of-line (EOL) mark
02500 ;
02510 EOL     CALL    BUMP1           ;XFER line separator
02520 REL19   EQU     $-2
02530 EOL1    PUSH    IY              ;Move IY address to
```

Listing continued

Listing continued

```
02540          POP     DE              ;  DE registers
02550          POP     HL              ;Recover line link address
02560          LD      (HL),E          ;Set LSB of link
02570          INC     HL              ;Bump pointer
02580          LD      (HL),D          ;Set MSB of link
02590          JP      LOOP1           ;Process next line
02600 REL2B    EQU     $-2
02610 ;
02620 ;  End-of-program processing
02630 ;
02640 DONE     PUSH    IY              ;Transfer address
02650          POP     DE              ;  to DE
02660          POP     IY              ;Discard old link addr.
02670          LD      HL,VAR_TBL      ;HL==> var. table storage
02680          STORE
02690          LD      HL,ARR_TBL      ;HL==> array table storage
02700          STORE                   ;Store address there
02710          LD      HL,FRE_SPC      ;HL==> free space storage
02720          STORE                   ;Store address there
02730 ;
02740 OUT      POP     IY              ;Recover registers
02750          POP     IX
02760          POP     HL
02770          POP     DE
02780          POP     BC
02790          POP     AF
02800          OR      1               ;Set NZ flag
02810          LD      A,0             ;Return null key
02820          RET                     ;Return to Basic
02830 ;
02840 ;  Bump and Transfer subroutine
02850 ;
02860 BUMP1    LD      B,1             ;Entry for single transfer
02870 BUMP     LD      A,(IX)          ;Get a byte
02880          LD      (IY),A          ;And store at new address
02890          INC     IX              ;Increment pointers
02900          INC     IY
02910          DJNZ    BUMP            ;Repeat until done
02920          RET
02930 ;
02940 FLTEND   EQU     $-1             ;End of filter
02950 FLTLEN   EQU     $-BEGIN         ;Length of memory-resident module
02960 ;
02970 ;---------------------------------------------------------------
02980 ;
02990 ;   Initialization code
03000 ;
03010 ;---------------------------------------------------------------
03020 ;
03030 INIT     PUSH    DE              ;Save DCB pointer
03040          LD      (MODDCB),DE     ;Stuff into filter
03050          LD      HL,SGNON        ;HL==> sign-on message
03060          SVC     @DSPLY          ;Display on screen
03070          LD      DE,MODNAME      ;DE==> module name
03080          SVC     @GTMOD          ;Already installed?
03090          JR      NZ,VIASET       ;Go if not found
03100          LD      HL,INSTLD       ;HL==> error message
03110 ERR_OUT  SVC     @DSPLY          ;Display the message
03120          LD      HL,-1           ;Set extended error
03130          SVC     @EXIT           ;And leave
03140 ;
03150 ;  Installed with SET command?
03160 ;
03170 VIASET   SVC     @FLAGS          ;Point IY to flags
03180          BIT     3,(IY+'C'-'A')  ;Test bit 3 of C-flag
03190          JR      NZ,SETHI        ;Go if SET used
03200          LD      HL,NOSET        ;HL==> error message
03210          JR      ERR_OUT         ;And leave
03220 ;
03230 ;  Reset HIGH$ and prepare to relocate filter
03240 ;
03250 SETHI    LD      HL,0            ;Function: get current value
03260          LD      B,L             ;B=0 >> select HIGH$
03270          SVC     @HIGH$          ;Get current HIGH$ value
03280          LD      (OLDHI),HL      ;Save old HIGH$
03290          JR      Z,RELOC         ;Go if no error
03300          LD      HL,MEMERR       ;HL==> error message
03310          JR      ERR_OUT         ;And leave
03320 ;
03330 ;  Move filter to high memory and protect
03340 ;
03350 RELOC    LD      IY,RELTAB       ;IY==> Relocation table
03360          LD      DE,FLTEND       ;DE==> End of filter
03370          XOR     A               ;Reset carry flag
03380          SBC     HL,DE           ;Calculate distance to move
03390          PUSH    HL              ;  and transfer to
03400          POP     BC              ;  BC register pair
03410 RELOC1   LD      L,(IY+0)        ;Get address to change
03420          LD      H,(IY+1)        ;  in HL
03430          LD      A,H             ;Pick up MSB
03440          OR      A               ;Is it 0?
03450          JR      Z,MOVE          ;Yes -- go move filter
03460          LD      E,(HL)          ;Move contents
03470          INC     HL              ;  of address
03480          LD      D,(HL)          ;  to DE reg. pair
03490          EX      DE,HL           ;HL has value to change
03500          ADD     HL,BC           ;Add the offset
03510          EX      DE,HL           ;New value back in DE
```

Listing continued

Basic. If you invoke the compression utility in that situation, it might run rampant trying to compress a /CMD program, garbage in memory, or even itself, and cause your computer to crash completely. Be careful!

The program begins to operate at line 1060. Since it's a keyboard filter, it must first call the keyboard driver routine to collect a keystroke. Then it compares that key to the constant signal to see if you're invoking it. If so, control passes to line 1180 where Squeeze pushes all the Z80 registers onto the stack and checks Basic's "signature." If everything is okay, compression begins at line 1490.

Throughout the program, the IX register points to the current location in the uncompressed code, and the IY register points to the current location in the compressed code. The outer program loop, which begins at LOOP1 in line 1560, executes for each line of your program. The inner loop, beginning at LOOP2 on line 1780, executes for each byte of the original program. The inner loop cannot just discard all spaces and remarks because the internal representation of numbers might contain bytes that look like spaces or remark tokens. Instead, the inner loop must copy all numbers completely, along with their tokens, and look only for bytes to discard between numbers and outside of literal strings' quotation marks.

At the end of the program (see line 2640), three of Basic's pointers need to be updated. If not, you will have a compressed program but no extra free memory because Basic still reserves memory space for your original program.

The program code following line 2920 is only to relocate and install the compression program. This is the same installation routine I've used many times and should look familiar to regular readers. The comments in the program should make most of it easy to follow.

I've used the program without problem on several Basic programs, and can usually reduce the size of a program by 25 percent or more. However, the three "apparently unused" entries in the Table bother me. They might be used for numeric types I have overlooked. If you find a program line that chokes the compression program consistently, please send it to me. You might have found a numeric token that I have overlooked, and I would like to add it to the list and publish a program patch. ■

You can contact Hordin Brothers through CompuServe. Go PCS-117 to the Writers' and Editors' SIG (WESIG). You can also write to Hordin at 280 N. Campus Ave., Upland, CA 91786. Enclose a stamped, self-addressed envelope if you want a reply.

Listing continued

```
03520           LD      (HL),D          ;Put it back
03530           DEC     HL              ;  in the
03540           LD      (HL),E          ;  filter program
03550           INC     IY              ;Bump IY to next
03560           INC     IY              ;  entry in the table
03570           JR      RELOC1          ;Repeat until done
03580 ;
03590 ;  Move module to high memory and protect
03600 ;
03610 MOVE      LD      DE,(OLDHI)      ;DE==> destination address
03620           LD      HL,FLTEND       ;HL==> current end of filter
03630           LD      BC,FLTLEN       ;BC==> length of module
03640           LDDR                    ;Move it all
03650           EX      DE,HL           ;Move new HIGH$ to HL
03660           LD      B,0             ;Function: select HIGH$
03670           SVC     @HIGH$          ;Set new HIGH$ value
03680           INC     HL              ;HL==> filter entry point
03690 ;
03700 ;  Set type and address in filter's DCB
03710 ;
03720           POP     IX              ;Get DCB address off stack
03730           LD      (IX),01000101B  ;Set as FILTER capable of
03740                                   ;  @GET & @CTL
03750           LD      (IX+1),L        ;LSB of filter address
03760           LD      (IX+2),H        ;MSB of filter address
03770           LD      HL,SUCCESS      ;HL==> Success message
03780           SVC     @DSPLY
03790           LD      HL,0            ;Show success
03800           SVC     @EXIT           ;Back to TRSDOS
03810 ;
03820 ;  Initialization messages
03830 ;
03840 MODNAME DEFM      'SQUEEZE'
03850           DB      0               ;Our filter's module name
03860 SGNON     DEFM    'BASIC Program Compression Utility'
03870           DB      CR
03880 INSTLD    DEFM    'Program already in memory -- installation aborted'
03890           DB      CR
03900 NOSET     DEFM    'Filter must be installed with SET command'
03910           DB      CR
03920 MEMERR    DEFM    'High memory not available for installation'
03930           DB      CR
03940 SUCCESS DEFM      'Installation successfully completed'
03950           DB      LF
03960           DEFM    'Use FILTER command to connect to *KI'
03970           DB      LF
03980           DEFM    'Then use <Clear><Shift><S> to invoke'
03990           DB      LF
04000           DEFM    'WARNING:  Do not invoke unless Basic is Active!'
04010           DB      LF
04020           DB      CR
04030 ;
04040 ;  Relocation table
04050 ;
04060 RELTAB    DEFW    REL1,REL2,REL3,REL4,REL5,REL6,REL7
04070           DEFW    REL8,REL9,REL10,REL11,REL12,REL13,REL14
04080           DEFW    REL15,REL16,REL17,REL18,REL19,REL20
04090           DEFW    0               ;Mark end with 2 bytes of 0
04100 ;
04110           END     INIT
```

End

On Displays: Sprucing Up Your Spreadsheet

Multiplan's Lookup function can give you a great deal of flexibility when you're doing calculations that must account for changing values. It will automatically go to a table, find the numbers that are right for the specified condition, and adjust its calculations accordingly.

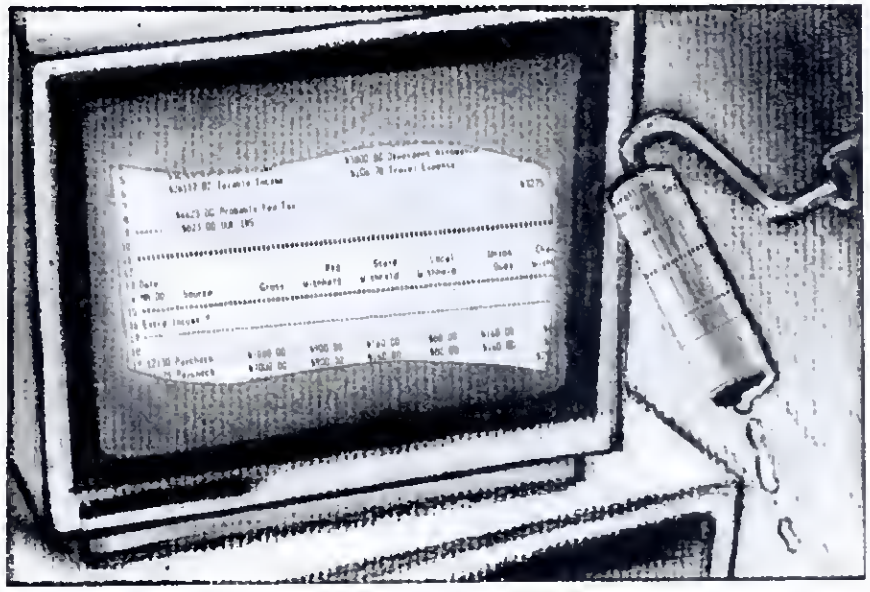
Let's look, for example, at a spreadsheet that calculates employees' income taxes (Fig. 1). The spreadsheet is for 1984, and assumes that the employees are married and filing joint returns. The equation to calculate federal taxes (column 3) is simple: Base + Percentage* Over-amount. All three variables in the formula, however, change with the salary of the employee. How can one equation in column 3 take these changes into account?

This is where the look-up table comes in. This spreadsheet refers to three, in columns 5–7. The base comes from column 6, the percentage from column 7, and the over-amount by subtracting the minimum salary (column 5) from the actual (column 2). The tax equation becomes the Multiplan formula in Fig. 2.

Let's get a quick look at how the spreadsheet works, using an income of $25,000 as an example.

The function Lookup (N,Table) searches for the first value (N) in the first row or column of the area specified by Table. Multiplan searches down a square or vertical table and searches left to right through a horizontal table. Lotus uses @VLOOKUP and @HLOOKUP to do the same.

Lookup searches down Salary_table to find $29,000 in row 13. Since this is a



	1	2	3	4	5	6	7
1		Employee Information		:		Tax Tables	
2				:			
3	EMPLOYEE NAME	SALARY	FED TAX	:	MINIMUM	BASE $	PERCENT
4				:			
5	Swanson, Clarke E.	57,825.00	14,341.50	:	0	0	0%
6	Harrell, John B.	32,469.00	5,589.32	:	3,400	0	11%
7	Harrell, J.B.	25,000.00	3,565.00	:	5,500	231	12%
8	Harrell, Bonnie S.	10,000.00	819.00	:	7,600	483	14%
9	Harrell, J. Matthew	3,000.00	0.00	:	11,900	1,005	16%
10				:	16,000	1,741	18%
11				:	20,200	2,497	22%
12				:	24,600	3,465	25%
13				:	29,900	4,790	28%
14				:	35,200	6,274	33%
15				:	45,800	9,772	38%
16				:	60,000	15,168	42%
17				:	85,600	25,920	45%
18				:	109,400	36,630	49%
19				:	162,400	62,600	50%
20				:	999,999		

Figure 1. Spreadsheet for calculating federal taxes using look-up tables.

Multiplan formula in column 3:
LOOKUP (RC[– 1],Base Table)
+ LOOKUP(RC[– 1],Percentage_table)*(RC[– 1] – LOOKUP (RC[– 1],Salary_table))

Base_table, Percentage_table, and Salary_table are named ranges of the tax table above as follows:
Salary_table—R3:18C5:5
Base_table—R3:18C5:6
Percentage_table—R3:18C5:7

Figure 2. Formula for spreadsheet.

Value	Color	Value	Color
0	Black	8	Gray (black on 2000)
1	Blue	9	Bright blue
2	Green	A	Bright green
3	Cyan	B	Bright cyan
4	Red	C	Bright red
5	Magenta	D	Bright magenta
6	Brown (yellow on 2000)	E	Bright yellow
7	White	F	Bright white

Figure 3. Color selections for Lotus.

Illustration by Jamie Hogan

"I am amazed at the broad spectrum of technical articles you publish, for both novice and advanced programmers . . ."

If you're still wondering which magazine you should buy for your TRS-80*, here's what **80 Micro** readers have to say about their #1 system-specific information source—

●"Not only is the magazine very professionally done, but I have found something in almost every issue that has been worth the price of the subscription . . ."

Roger L. Holstege
Millersville, MD

●"I was greatly impressed by your magazine. I got more useful information from that one issue of 80 than I have from countless other sources . . ."

John M. Crittenden
Jackson, MS

●"I have found **80 Micro** to be the most valuable magazine pertaining to home computers on the market . . ."

William C. Hardin, Jr.
Charlotte, NC

80 Micro is the magazine for every TRS-80 user—from beginner to advanced. **80 Micro** is full of tutorials, free programs, hardware modifications, new product announcements, product reviews, debugging tips, and more.

And an **80 Micro** subscription is risk-free. *If you're not completely satisfied, you'll be reimbursed for all undelivered issues.* See what **80 Micro** can do for yuu. It's #1 for a lot of people. Fill out this order form and send it in now.

*TRS-80 is a trademark of Radio Shack, a division of Tandy Corp

Yes! I want a no-risk subscription to the magazine for beginner to advanced TRS-80 users. Send me 12 issues of 80 MICRO for $24.97! I'll save 48% off the newsstand price!

☐ Payment Enclosed ☐ Bill me

Name _____

Address _____

City _____ State _____ Zip _____

Canada & Mexico, $27.97. Foreign surface, $44.97. 1 year only. US funds drawn on US bank. Please allow 6-8 weeks for delivery.

80micro • PO Box 981 • Farmingdale, NY 11737

35DB8

single-column table, the value returned is from the preceding row—$24,600. Lookup then searches down Base_table and Percentage_table, again stopping at row 12. The base, then, is $3,465, and the over-amount is $400 ($25,000 – $24,600). Plugging the numbers into our equation, the federal tax is $3,465 plus 25 percent of $400, or $3,565.

Put all of this together, and you have a nice tax calculator. If you let your imagination wander, you quickly realize that the formulas for tax calculations in column 3 could be extended to use any of four table areas, depending on whether you are married or single.

Action	Comments
DEBUG TD.DRV <ENTER>	Load Debug and the LOTUS text display video driver.
E17D <ENTER>	Begin editing memory at offset address 017DH in the driver.
WW <SPACE>	Enter first color selection from Fig. 3; enter the hexadecimal digits from the color selection table in the proper order for each attribute byte.
XX <SPACE>	Enter second color selection.
YY <SPACE>	Enter third color selection.
ZZ <ENTER>	Enter fourth color selection.
W <ENTER>	Write the modified TD.DRV driver back to the disk.
Q <ENTER>	Exit Debug to MS-DOS.

Figure 4. Instructions for modifying colors in IBM-PC Lotus.

Color Byte	IBM PC Lotus	Tandy 2000 Lotus
WW	Used to set normal colors.	Used to set the background colors for normal text and borders.
XX	Used to set colors for the spreadsheet border.	Used to set the foreground color for normal text and borders.
YY	Used to set colors for unprotected cells and help text without the cursor.	Used to set the background color for help text and unprotected cells.
ZZ	Used to set colors for unprotected cells and help text with the cursor.	Used to set the foreground color for help text and unprotected cells.

Figure 5. Description of bytes used to modify Lotus colors.

Action	Comments
DEBUG TD.DRV <ENTER>	Load Debug and the LOTUS text display video driver.
E18D <ENTER>	Begin editing memory at offset address 018DH in the driver.
WW <SPACE>	Enter first color selection from Fig. 3; enter the hexadecimal digits from the color selection table in the proper order for each attribute byte.
XX <SPACE>	Enter second color selection.
YY <SPACE>	Enter third color selection.
ZZ <ENTER>	Enter fourth color selection.
W <ENTER>	Write the modified TD.DRV driver back to the disk.
Q <ENTER>	Exit Debug to MS-DOS.

Figure 6. Instructions for modifying colors in Tandy 2000 Lotus.

Circle 470 on Reader Service card.

96K MODEL 100

The *PG Design* 64K RAM module adds two additional 32K RAM banks to your existing Model 100. The firmware that comes in your Model 100, BASIC, TEXT, TELCOM, ADDRSS, SCHEDL, are all present in each additional RAM bank.

You may transfer files from one RAM bank to any of the other RAM banks by using the function keys.

We designed our RAM module to allow clear access to the ROM slot and we provided a forty pin, male connector to utilize Radio Shack's DVI or other disk drive systems. We have installed a lithium power cell to maintain the memory of the module for six months if you should remove it from your Model 100. But, while in the Model 100, the power cell has a life span over six years!

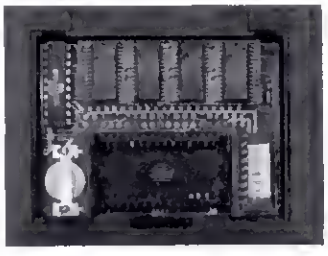
The RAM module snaps easily into the expansion port in the bottom of your Model 100.

Guarantee

We guarantee that all PG Design products perform to your complete satisfaction, or your money back! We manufacture the highest quality RAM modules available anywhere.

Order Today	
64K RAM module	$375
32K RAM version	$250
Upgrade later for	$150
*8K Ram modules for Model 100— Set of three—	$29.95 ea $84.00
*24K RAM modules for Model 200— Two or more—	$139 $135 ea.

send check, money order, Visa, & Mastercard accepted



PG Design Electronics, Inc. 66040 Gratiot Richmond MI 48062 (313)727-2744

Circle 452 on Reader Service card.

FX/RX Owners

THIS (Draft Print)  or THIS (NLO Print) (5 x size)  ?

Install our new LetterWriter NLQ upgrade kit in your printer and you can choose NLQ (or any of 15 other print features) by simply tapping your printer panel buttons. NLQ print has four times the resolution of draft print. That means finer character detail, no visible dots, and impressive-looking documents. The FX version even adds IBM Graphics printer compatibility. With LetterWriter in your FX or RX, you may never need to buy a letter quality printer. Why not get LetterWriter for your printer today?

NLQ 12345abcdefghiJKLMNOPQRS
Draft 12345abcdefghiJkLMNOPQRS

FINGER PRINT™ LetterWriter™ NLQ Upgrade Kit by:

$79.95 FX $59.95 RX

For all FX-80/100 and RX-80/100 printers including F/T and + series
Simple plug-in installation
Complete instructions included

Dresselhaus Computer Products, Glendora, CA
Call (818) 914-5831 for dealer nearest you.

Circle 536 on Reader Service card.

FastPak Business Mail System

Correspondence
Direct Mail
Mailing Lists
Addressing
Form Letters
Tickler Files
Newsletters
Sales Letters
Credit Letters
Sales Follow Up
Collection Letters
Boilerplating
Envelopes
Invoicing
Labels
Notices
Resumes
1099 Forms
Forms Fill-in

Premium Software / Affordable Prices

Any place you need a name and address

FastPak Mail from DHA Systems & Software. A complete library of programs that does everything for mail but lick the stamps. No frustration, no learning, no set-up. Easy to use.

Organize all your name, address, and phone files in one place. Input and corrections are easy, just fill in the blanks. Add your own codes. Fast, easy, flexible sort. Select names for special mailings.

Run labels and letters at the push of a button.

FastPak Mail includes a powerful system for merging letters and mail files for any kind of letter writing — from routine correspondence to personalized direct mail letters.

"Your product is excellent . . ." — *John Stevenson, Experts in Direct Marketing*

". . . the best direct mail product aid I ever bought." — *Ralph Thomas, Thomas Publications*

"FastPak Mail is a bargain to say the very least."

"We are totally amazed at what your mailing list system can do. The sort and merge functions are fantastic, as is the entire package in general." — *Steven Friedman, SHF Software Systems.*

DHA Systems & Software

832 JURY COURT / SAN JOSE, CA 95112 / (408) 947-1011

Merge - Works with all the popular word processors, Word, Wordstar, Multimate, etc. Create form letters with fill-ins. Easy enough for 1 letter, powerful enough for 5000.

Sort, Select, Combine - Lets you organize your files anyway you want, by zip, names, or special codes.

Conversion - no need to retype your list. Easily convert your existing lists to our format.

Purge - Eliminate duplicate names

Economical Easy to buy **$79.95**

To Order: Send check or money order for $79.95 plus $5.00 shipping and handling. California residents add sales tax. Phone orders, Call 800/FastPak. Visa/MC welcome.

RUNS ON MODEL 4 AND MODEL 1000 (CPM OR MS-DOS)

Lotus Land

I get bored with the color selections on the IBM and Tandy 2000 versions of Lotus 1-2-3. Color displays are meant to display color! Changing your Lotus colors takes less than five minutes and anyone can do it.

Let's tackle the IBM-PC compatibles first. Lotus sets the colors for each character by writing a distinct color attribute for each position. Color video memory is organized into pages of 4,000 bytes containing a character byte followed by an attribute byte.

The attribute byte contains two 4-bit numbers that identify the color of the character background and the color of the character itself. The foreground color can be any of the colors in Fig. 3 while the background color is limited to selections zero—7. Adding eight to the background color forces the character to blink. For example, 1F gives bright white characters on a blue background, while 9F causes the characters to blink.

Now, armed with this knowledge, select background and foreground colors for the spreadsheet border, the text on the spreadsheet, unprotected cells or unselected cursor locations in the help mode, and unprotected cells or actual cursor location in the help mode. Write these down, remembering to organize them in each byte as background/foreground, and fire up Debug.

Place a disk containing Debug in drive B and the Lotus system disk (it must contain the file TD.DRV) in drive A. Follow the instructions in Fig. 4, entering each step just as it appears. Replace WW, XX, YY, and ZZ with your color values; see Fig. 5 for a description of each byte. When you're done, run Lotus Access; you should see the changes immediately.

The Tandy 2000 Lotus works differently on the 2000 than it does on other MS-DOS machines. The spreadsheet frame and text are displayed using some colors in the monochrome text mode.

For the Tandy 2000, the monochrome video is organized into a single page of text arranged like the IBM PC. Unfortunately, the attribute bytes don't resemble the PC's. These attributes do allow setting normal or high-intensity display, blink, underlining, and reverse video. The normal and high-intensity modes select their respective colors from the palette register contents and you can control these values. All 16 colors in Fig. 3 are allowed for setting the values.

Normal or highlight characters may also be displayed in reverse video. Lotus uses these four combinations to display all text. While you have no control over the attributes used to display text, you can control the colors used for each mode.

Again, use Fig. 3 to select the colors you want. The first will control the background color of the normal text and the second the foreground color. These colors will also be used for the border, which is displayed in reverse video (the functions of these two colors are reversed). The third and fourth choices set the colors for the help text and unprotected cells. You enter each of these colors as a single byte; for example, bright green as byte 0A.

Figure 6 gives directions on using Debug for the Tandy 2000. Follow them as you would those for the PC-compatibles. You should immediately see your color selections when you run Lotus.

A final word of warning: You can select some bizarre color choices. You might need to experiment before you find the colors you like. ■

Write to Spreadsheet Beat c/o 80 Micro, 80 Pine St., Peterborough, NH 03458. We will pay $50 for any templates that we publish.

Circle 276 on Reader Service card.

$395 Complete

Radio Shack
TAX SOFTWARE

TAX/PACK

ACCOUNTANT & PRACTITIONER SYSTEM

Runs On TRS 80 MOD 3-4 On CP/M
TRS 1000-1100-2000 On MSDOS
TRS 80 MOD. 2-12-16 On
TRSDOS Or CP/M

• MORE FORMS & SCHEDULES
Tax/Pack processes all schedules and forms 1040, 2106, 2119, 2210, 2441, 3468, 3800, 3903, 4562, 4797, 4815, 4897, 5695, 6251 and 6252. The system also includes client billing & cover letter attachment, depreciation and mortgage loan schedules. Batch print module available.

• INSURES SPEED & ACCURACY
Enter data only once. The data is passed to appropriate forms and schedules automatically for speedy processing and changes. Or up to 5 Schedule C's, D's, E's, F's, etc. per client.

• FULL SUPPORT & UPDATE
Delivery by January 15. Receive a free update disk in February. Phone modem support. Annual update fee only $125. Available source code version lets user customize programs and output.

• RECEIVE A FREE '84 TAX/PACK
Order the 1985 version before Nov. 15 and receive a free '84 version for employee training. Demo disk only $25.00. Personal Tax Pack $79.95. Client Proforma and Worksheet Generator also available.

Many New Features Added

ALPINE DATA

635 MAIN MONTROSE, CO 81401
(303) 249-1400 **1-800-525-1040**

MasterCard VISA

TAX SOFTWARE SINCE 1978

Radio Shack Registered Name Of Tendy Corp.

Circle 185 on Reader Service card

70 INCOME TAX PROGRAMS

(For Filing by April 15th, 1986)

TRS-80 Models I, II, III, 4/4P, 12 and 16 and this year: PC-DOS and MS-DOS*

FEATURES: —

1. *Our 7th year in TAX Programming*
2. *Menu Driven Programs*
3. *"SAVE" on Disk*
4. *View on Screen before Printing*
5. *Correctable*
6. *BASIC, Unprotected*
7. *Don't change paper ALL SEASON!*
8. *We Stand Behind our Programs!*
9. *Write for Details.*

For the Tax Preparer, C.P.A., Lawyer and Individual. Hundreds of long-time Users. You buy ONLY the disks you need.

Programmed for easy-use. Programs follow the Forms closely. Check-points along the way. Results on screen before printing; can be corrected.

70 TAX PROGRAMS include: Forms 1040, 1040A, 1120, 1120A, 1120S, 1065 and 1041. Also ALL Schedules, and Forms 1116, 2555, 2106, 2119, 2210, 2441, 3468, 3800, 3903, 4136, 4137, 4255, 4562, 4684, 4972, 4797, 5695, 5884, 6251, 6252, 6765, 8027, 8283, 8332 and 8379. Also Tax Preparer HELPER disks.

There are 14 disks (@ $24.75) for the Model I; 7 disks (@ $49.50) for the Models III, 4/4P, PC-DOS and MS-DOS; and 3 disks (@ $99.) for the Models II, 12 and 16. Buy only the disks you'll need. MO & ME buyers, add Sales Tax, please. Write for details.

MasterCard VISA

For TRSDOS Programs, WRITE: —
GOOTH TAX PROGRAMS
931 S. Bemiston • St. Louis, MO 63105

***For PC-DOS and MS-DOS Programs, write:**
MICROCRON SYSTEMS
P.O. Box 561, Old Town, ME 04468

Circle 71 on Reader Service card

T80-FS1 Flight Simulator

Available for Model I or Model III. $25.00 on cassette or $33.50 on disk (with enhancements) All versions require 16K.

If you order direct, please specify whether you have Model I or Model III (the media are different) and whether you want disk or cassette. Include $1.50 and indicate UPS or first class mail. Illinois residents add 5% sales tax. Visa and Mastercard accepted.

If you don't yet own a disk, don't fret. You can upgrade anytime. Cassette users may send back their cassette (but not the manual) along with $10 (first class shipping included) and receive the disk version.

See your dealer!

subLOGIC

Communications Corp.
713 Edgebrook Drive
Champaign, IL 61820
(217) 359-8482
Telex: 206995

$777

5 MEG HARD DRIVE

NEW!

Comes complete with cable and choice of software driver* (CPM, LDOS, TRSDOS)

10 Megabyte Hard Drive	$899
15 Megabyte Hard Drive	1095
30 Megabyte Hard Drive	1495

SYSTEM FEATURES

- For TRS-80 Model 3-4-4P
- One Year Parts & Labor Warranty
- Size Rated After Format
- Continuous Duty Power Supply
- Error Checking & Correcting
- Continuous Duty Fan
- Size 11.5" x 12.5" x 5.0"
- All Contacts Gold Plated

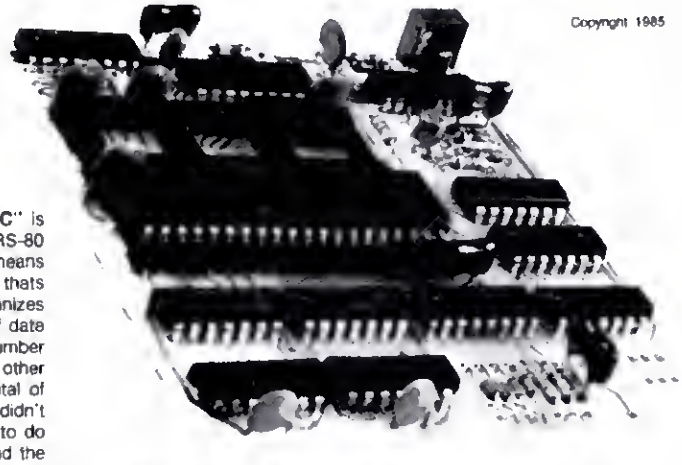
*Model 3 requires LDOS
Model 4 requires TRSDOS 6.2 or Montezuma Micro CP/M 2.2

There are firms which offer benefits, experience or products seemingly too good to be true. Now why would you want to expose yourself to unhappiness when Aerocomp has a proven record of thousands of happy, satisfied TRS-80 customers. Just take a minute to look through back issues of this magazine. You won't find many companies that have been around as long as Aerocomp. We fully support TRS-80 computers and most all operating systems including CP/M 2.2. Aerocomp leads the way to low hard disk prices so you can afford to enjoy the benefits of increased storage and faster disk I/O. These units are precision engineered, tested and delivered complete and ready to use, right from our stock. Each unit is guaranteed for one year parts and labor. You can count on us to be here if you should ever need us. As always, your satisfaction is assured with our 14 day free trial offer. If, for some reason, you are dissatified with our drive merely return it for a full refund (less shipping). How can you go wrong? Specify the software driver of your choice and start enjoying your computer's real capability. Do it today! Call our toll-free number now!

See opposite page ▸▸▸▸▸▸

MODEL I DOUBLE DENSITY BOARD

Add 80% more capacity to your disk system with the Double Density Controller (DDC) from Aerocomp.



The Story

Some products have what it takes to seem to last forever. Our "DDC" is one of those products. What it does is allow you to operate your TRS-80 Model 1 disk system in double density. In this case double density means almost doubling the storege capacity of your diskettes. Single density, thats the way Radio Shack designed your Model I expansion interface, organizes your disk into 10 sectors per track. Each sector conteins 256 bytes of date for e total storage capacity of 2,560 bytes or 2.5K per track times the number of tracks your drive is capable of adressing. Double density, on the other hand, writes 18 sectors per track eech containing 256 bytes for a total of 4,608 bytes or 4.5K. That is 80% more data in the same space. Why didn't Radio Shack do that in the beginning, you ask? Well it costs money to do double density because it is more difficult to do than single density and the data is harder to capture reliably. That means more cost and the Model I was meant to be a low-cost computer for the masses. Therefore, no double density for the original Model I.

The Facts

Other companies introduced double density controllers for the Model I but they were not so good. We waited and waited but, even new models failed to correct problems with data seperation that kept cropping up. So we went to work and came up with a new design to cure the old problem. At last! A double density controller for the Model I with a higher probability of data recovery than with any other double density controller on the market then or since. Our analog design phase lock loop data separator has a wider capture window than the digital types the others use. This allows high resolution date centering. Our "DDC" analog circuit allows infinately variable tuning. The attack and settling times are optimum for 5.25" diskettes. The oft-stated fears of adjustment problems rumored by digital dilettentes have been proved groundless by thousands of satisfied users the world over. The bottom line here is state-of-the-art performance and reliability.

ORDER YOURS TODAY

TRS-80 Model I disk system owners who are ready for reliable double density operation will get 80% more storage per diskette; single and double density data separation with far fewer disk I/O errors; single density compatibility; simple plug-in operation. You will need a disk operating system that has the necessary double density software. All the popular DOS's (except TRSDOS) have the drivers. We have put together two special packages in the event you don't already have one of the more popular DOS's.

Please add $4 handling & shipping

"DDC" by itself	$ 99
"DDC" and LDOS	159
"DDC" and NEWDOS 80 2.0	179

☞40 Track
☞80 Track
☞Single Head
☞Dual Head
☞Bare
☞Complete
☞Full Size
☞Half Size
☞3-1/2", 5-1/4" or 8"

DISK DRIVES

Aerocomp leads the way to the BEST value in disk drives and related peripheral products on the market today. Sound engineering, high performance, quality construction, no-risk free trial, outstanding warranty service and a reputation for doing the right thing make your decision to buy Aerocomp the correct one. Please look over our offerings and make your selection. When you have made your choice call our toll-free number and place your order. If you need assistance in making your selection, please call our information number. It's listed in the box at the bottom of this ad along with the technical assistance number for those of you who want to get right to the nitty-gritty. Thanks, we all appreciate your business and will continue to do our very best to support you.

BARE DRIVES Add $4 S&H

These drives are completely burned-in and tested for your ultimate satisfaction. Add that extra drive today! These are new factory drives. No blems, seconds, closeouts or defunct manufacturers surplus (MPI, Shugart, etc). Instruction manual included at no extra cost. Service manuals are also available. See "Miscellaneous Goodies" for information.

40tk SS half high TEAC FD35-A ..... $ 99
40tk DS half high TEAC FD35-B 109
80tk DS half high TEAC FD35-F 129
40tk SS full size Tandon TM100-1 119
40tk DS full size Tandon TM100-2 119
40tk SS half high TEAC FD55-A 89
40tk DS half high TEAC FD55-B 99
80tk DS half high TEAC FD55-F 119
8" SS Thinline Tandon TM848-1E 260
8" DS Thinline Tandon TM848-2E 335

8" FLOPPY DRIVES Add $12 S&H

These 8" Thinline drives work with the Model 2 and 18 plus others that use standard drives. The rugged all-steel cabinet has an extra heavy duty power supply rated for continuous operation. A removeable air filter allows only clean air to circulate. Cabinets with single drives are supplied with a blank panel to cover the unused space. A second drive can be added at any time. Service manuals are available. See "Miscelleneous Goodies" for info.

1-8" SS Tandon TM848-1E & case ..... $ 389
2-8" SS Tandon TM848-1E's & case 649
1-8" DS Tandon TM848-1E & case 439
2-8" DS Tandon TM848-2E's & case 699

MISCELLANEOUS GOODIES $2 S&H

Model 1 TRSDOS 2.3 disk & manual ..... $ 25
Model 3 TRSDOS 1.3 disk & manual 25
Model 4 TRSDOS 6.2.x disk & manual 20
LDOS (specify Model I or III) 69
NEWDOS 80 v2.0 (specify Model I or III) 99
Montezuma Micro Model 4 CP/M 2.2 169
Tandon TM100-1/2 Service Manual 20
Tandon TM848-1/2 Service Manual 20
TEAC FD55A/B/F Service Manual 20
10 SSDD disks in library box, Lifetime Guar 12
10 DSDD disks in library box, Lifetime Guar 15
5.25" drive power supply & enclosure 59
8" drive pwr sply & encl w/fan 5V-24V 150
5.25" 1-drive cable, a nice length 19
5.25" 2-drive cable, a 44 inches 24
5.25" 3-drive cable, just the right size 29
5.25" 4-drive cable, the top dog 34
5.25" Extender cable with gold contacts 10

COMPLETE DRIVES Add $6 S&H

Fits TRS-80 Models 1,3,4 and the Color Computer plus the others that use standard drives. The drive of your choice is mounted in a sturdy, all-steel cabinet. An external gold plated drive connector allows cabling without disassembly of the cabinet. Half-high drives come in a full-size cabinet that will hold and power our half-high drives. Single drives have a panel covering the unused space allowing a second drive to be added at any time. All are shipped fully assembled ready to use. Specify silver or beige cover. Stainless Steel covers are available for $12 additional.

1-40tk DS 3.5" FD-35B in dual case ..... $ 159
2-40tk DS 3.5" FD-35B's in dual case 259
1-80tk DS 3.5" FD-35F in dual case 179
2-80tk DS 3.5" FD-35F's in dual case 289

40tk Single Side full size TM100-1 149
40tk Double Side full size TM100-2 159

1-40tk SS half-high FD-55A in dual case 129
2-40tk SS half-high FD-55A's in dual case 219
1-40tk DS half-high FD-55B in dual case 139
2-40tk DS half-high FD-55B's in dual case 259
1-80tk DS half-high FD-55F in dual case 159
2-80tk DS half-high FD-55F's in dual case 279

FREE TRIAL OFFER

Use your Aerocomp hardware product for up to 14 days. If you are not satisfied for **ANY REASON** (except misuse, damage or improper handling), return it (insured) in the original shipping container for a full purchase price refund, less shipping. **Sorry, this offer does not apply to software. Defective software will be replaced.** Any hardware/software specials will be prorated and the software will be charged at the regular unbundled price. We have confidence in our products and we know you will be satisfied.

WARRANTY

We offer a one year warranty on parts and labor against defects in materials and workmanship. In the event service becomes necessary for any reason you will find our service department fast, friendly and cooperative. We want to keep you happy. Out of warranty repairs are also available.

100% BURN-IN and TEST

All our products are burned-in and fully tested prior to shipment. We want you to receive an item ready-to-go. AEROCOMP means reliability!

ORDER NOW!

Call our toll-free number service and place your order. Have your American Express, Mastercharge or Visa number ready. We will not charge your card until the day we ship your order. You may order by mail using your credit card, check or money order. Personal and company checks are welcome and cause no shipping delay as long as they are bank printed and the signature exactly agrees with the name printed on the check. We will ship surface COD with no deposit but all COD's require cash or a cashier's check on delivery. Texas residents add 6% State Sales Tax. No tax collected on out of state shipments. Canadian addresses add $20 to your order if over $550 for customs documentation.

TRS-80 Model III & 4 DISK CONTROLLER and DRIVE KITS

Convert your cassette Model III or 4 to disk operation with one of our complete kits. You receive our own advanced disk controller board with gold plated edge contacts capable of 4-drive operation; our own power supply; plated steel mounting towers complete with RFI shield plus all the cables and hardware necessary. Detailed instructions are included. All you need is a screwdriver and a pair of pliers. System kits come with 40 track single-side drives or just order the basic kit and pick the drives you want from the selection in the next column.

CONTROLLER KIT ..... **$ 199**
(Everything you need - less drives and DOS)
1 DRIVE SYSTEM **299**
2 DRIVE SYSTEM **399**
Substitute DS drives for only $10 each.
Add $9 shipping & handling

Model 3 TRSDOS 1.3 disk/manual 25
Model 4 TRSDOS 6.2.x disk/manual 20
LDOS 5.1.4disk/manual 69
NEWDOS 80 version 2.0disk/manual 99
CP/M 2.2 by Montezuma Micro 189
No S & H charge when ordered with kit
MOUNTING KIT AND POWER SUPPLY ONLY **95**
Add $8 shipping and handling
DISK CONTROLLER ONLY **110**
RS-232 BOARD & KIT **69**
Add $4 shipping and handling

OUR FAMOUS MODEL I STARTER PACKAGE

If you have a Model I and an Expansion Interface this is what you need to get started with disks. Included is one 40 track single-side disk drive complete with matching silver case and power supply, a 2-drive cable, a TRSDOS 2.3 disk operating system and TRSDOS manual plus all insurance and delivery charges to your door (lower 48 states).

Yours for only **$ 199**

LDOS 5.1.4 disk/manual 69
NEWDOS 80 version 2.0disk/manual 99

You can add our renowned "DDC" double density controller to either the Radio Shack or the LNW Expansion Interface for 80% more storage capacity on your drive. Order it at the same time as our starter package above and we'll pay the shipping. Go ahead, you deserve increased density. See the opposite page for the latest technical details.

$ 89
When purchased with Starter Package

CALL TOLL-FREE

800-527-3582 USA
800-442-1310 TEXAS

For inquiries or information
call 214-339-8324

AEROCOMP
Redbird Airport, Bldg. 8
P.O. Box 24829
Dallas, TX 75224

Continued from p. 34

Hyperzap also lets you inspect and modify memory with string searches, CRC calculations, memory modifications, block moves, block fills, and block comparisons.

Hyperzap generates self-booting disks for either a Model I, III, or 4. Also, you can put any combination of I, III, or 4 programs on a self-booting disk, useful for anyone distributing Model I and Model III/4 versions of a program on a single disk.

Hyperzap supports a feature called autopilot, a do-file mimic for building files of multiple keystrokes. Once you build an autopilot file, or use one of Hyperzap's 17 files, you can pass control to Hyperzap and all program prompts will be answered by the autopilot file.

Drawbacks

While all of the above may sound great, I found a number of serious problems with Hyperzap. The 39-page manual provides a good explanation of Hyperzap's many features, but it doesn't explain disk formats and structures.

Also, entering data with Hyperzap is a confusing hodgepodge. Many program prompts require a leading zero for single-digit numeric values. Others require only a single digit and produce incorrect results if you add the leading zero. While you can put most numeric responses in either decimal or hexadecimal format, some prompts accept only decimal or hexadecimal values. To aggravate the situation further, incorrect responses can sometimes crash the program.

Since Hyperzap runs independently of a disk operating system, it uses its own device driver routines. While the video and printer drivers function properly, the keyboard and disk drivers exhibit a few glitches. The keyboard driver doesn't provide enough debounce, so the keys repeat slowly. And the disk driver hangs up completely whenever you try to access a diskless drive.

My first copy of Hyperzap indicated I had numerous CRC errors on disks that I could format without problem with other operating systems. Hypersoft sent me another copy of Hyperzap that worked fine.

Conclusion

Hyperzap offers several unique features for a zap utility, but the program's problems make it difficult to use. If Hypersoft corrected Hyperzap's weaknesses, I would give the program much higher marks. But I found the inconsistent data entry requirements confusing. With a little refinement, Hyperzap would be an excellent product. Until then, I can only consider it fair.■

Multilingual MULTIDOS 80/64

by Thomas L. Quindry

★★★½

MULTIDOS 80/64 runs on the Model 4 (64K) and requires one disk drive. AlphaBit Communications Inc., 13349 Michigan Ave., Dearborn, MI 48126, 313-581-2896. $99.95.

Easy to use: ★★★☆☆
Good docs: ★★★☆☆
Bug free: ★★★★☆
Does the job: ★★★☆☆

As any of its fans know, MULTIDOS reads and writes practically any Model I/III DOS format. The Model 4 version of MULTIDOS, 80/64, extends this feature to include TRSDOS 6.X. MULTIDOS 80/64's main advantage, however, is its ability to run Model I/III Basic programs in Model 4 mode without conversion, at the Model 4's faster processing speed and 80-character screen width (unless the program uses machine-language subroutines or PEEKs or POKEs).

Like other versions of MULTIDOS, 80/64 touts its ability to read all disk formats for the TRS-80 series. While MULTIDOS can directly read some DOS formats (like LDOS), you have to use a program called VFU to convert TRSDOS 1.3 programs to another drive to run them. Several Model I DOSes require that you change the disk's data address marks with the MULTIDOS Convert/CMD program. As with other versions of MULTIDOS, 80/64 can read all TRS-80 disk formats. It also writes to most formats, but not to TRSDOS 1.3 and 2.3.

Compatibility

MULTIDOS Basic uses tokens identical to those in Model I/III Basic, but TRSDOS 6.X Basic uses different tokens. Therefore, you must save Model 4 Basic programs in ASCII format before MULTIDOS can read and run them directly from a TRSDOS 6.X disk.

MULTIDOS's Basic interpreter, SuperBasic, comes with enhancements to standard Basic and debugging tools. You can trace, single-step, set breakpoints, and review variables in Basic programs. You also get a string sort similar to that in TRSDOS 1.3 Basic, with output in ascending or descending order.

Additional Basic commands include Label, Erase, Zero, Hex, Binary, Call, and WPEEK. Erase removes a variable array from RAM. Zero sets all elements of the array to zero. WPEEK PEEKs at a 2-byte value (word) that an integer points to. Some of these SuperBasic commands conflict with those in TRSDOS 1.3.

MULTIDOS Features

While MULTIDOS 80/64's compatibility with Model I/III Basic programs is good, it is a Model 4 operating system. It resides in RAM and loads its Basic interpreter in low memory. Because of this, you can't run most /CMD files written for the I, III, or 4. MULTIDOS also doesn't support the RAM calls most commercial software packages use. For instance, I couldn't run Scripsit, LeScript, or Allwrite. Some commercial machine-language programs can access most features, but not all.

If you have a 128K Model 4, MULTIDOS lets you partition the extra memory bank as a Memdisk; you can also set aside part of high memory as a RAM disk or data disk. MULTIDOS provides a MINIDOS function accessible at all times, even while running a Basic program. It includes commands you can run before returning to the program. You can copy, kill, or list specified files; display a directory; invoke a debugging program; and select the 64- or 80-character screen widths (32 or 40 characters in enlarged-character mode).

Other useful commands available from DOS include an Unkill command and linking and routing commands. MULTIDOS's utilities let you assign function-key characteristics, edit globally in Basic, zap disks, time disk drives, filter printer codes, spool printer data, test memory, and scan/search memory for 8- or 16-bit codes.

MULTIDOS supports double-sided disk drives, but the manual provides no instructions for making a double-sided MULTIDOS system disk.

You can also format, read, and write to the reverse side of a double-sided disk as though it were a separate drive. Each side acts as an independent disk with its own directory; you refer to a two-drive system's four "drives" as 0, 0', 1, and 1'. However, you have to configure MULTIDOS to recognize double-sided disks.

Docs and Knocks

The MULTIDOS manual is tough to get through. While it offers good technical information, you have to hunt all over for it. And I found the way it handles the different systems confusing.

I discovered only one error with MULTIDOS. The Memdisk X command is supposed to reset the Memdisk or RAM disk previously set, but I couldn't get it to work.

Conclusion

While MULTIDOS 80/64 isn't fully compatible with Model I, III, or 4 programs, it does have some features you can't find anywhere else. MULTIDOS 80/64 has utility for a select audience and you may just be one of them.■

Typitall: A Scripsit Alternative

by David Dalton

★★★½

Typitall runs on the Models I, III (48K) and 4/4P (64K) and requires one disk drive. Howe Software, 14 Lexington Road, New City, NY 10956, 914-634-1821. $129.95. With spelling checker, $179.95.

Easy to use: ★★★★☆
Good docs: ★★★☆☆
Bug free: ★★★☆☆
Does the job: ★★★☆☆

If you're a Model III Scripsit user who has moved up to a Model 4, you'll probably like Typitall, an inexpensive and capable word processor.

While Typitall uses some of the Model 4's extra features, such as the 80-column screen and the function keys, it doesn't use the extra memory available with 128K systems. Under TRSDOS 6.X, Typitall holds only 41K of text. It also doesn't use the Model 4's reverse video to highlight text, as SuperScripsit does.

Features

Typitall adds some important features missing from Model I/III Scripsit. You can send special codes to the printer, for example, and execute DOS commands from within the program. You can even exit to DOS, do a few chores (such as formatting a disk), and return to Typitall with your text intact.

Typitall calls help files at the touch of a key, and updates a status line at the bottom of the screen after each keystroke. It displays the current line number, the length of the line, the line width, the document length, and the amount of free space in bytes.

Inserting new text within old was always a pain with Scripsit. Typitall makes it easier with the function keys. F1 opens a line for inserting text, F2 deletes one character, and F3 rejoins lines after an insert. You use control-M to switch back and forth between overstrike and insert mode.

You print files to the screen or to a disk file. Printing to the screen lets you check your format without wasting paper. You can also print to the screen using small graphics blocks instead of text, which will display how the pages will look.

One nice Typitall feature is its printer spooler. You can save a document to a disk file and have Typitall print the file while you work on something else. Typitall has some limitations here, though. It may ignore your keystrokes while it goes to the disk for the next block of text to print, and a noticeable system slowdown signifies that you're using the spooler. Printing from TRSDOS's Memdisk isn't as slow.

Customizing

You can modify many of the program's features and parameters and save them to disk permanently. You can, for example, change the rate at which keys repeat or set up a default file name extension.

You can also set up printer parameters, such as whether your printer expects line feeds, and send command strings to reset the printer each time you print a file. Typitall's printer support is good, but it doesn't support proportional spacing or serial printers. You can set up sequences of keystrokes and save them permanently. Thereafter, you can call often-used command routines or character strings with one keystroke. This is a good way to save printer-control lines that you use frequently.

To give you more room for your documents, Typitall uses several overlays. That means that only part of the program resides in memory at any one time. If you want to print a file, Typitall reads the printing overlay from disk, as it does the help files. You can circumvent this process by copying the overlays and help files to Memdisk and customizing Typitall so that it accesses Memdisk before loading an overlay. This makes things run faster.

Problems

Typitall did several weird things with my documents. I was unable to reproduce the problem, but a couple of times my screen width changed of its own accord and the text became skewed, though I lost none. Sometimes an invalid command will slightly alter the appearance of your text at the cursor location.

Spelling Checker

The spelling checker, which only costs an extra $50, contains about 29,000 words. It's slow, awkward to use, and the size of the document that it checks is limited by available memory. The checker sorts your document to make a list of unique words, looks up the words in the dictionary, and drops them into a block at the top of your file. You use a Hunt command to find each misspelled word in your document. You can add to the dictionary and create your own special dictionaries.

Conclusion

Typitall lacks the power of SuperScripsit, the pizzazz of LeScript, and the class of Allwrite. But not everyone can deal with Allwrite's price tag or SuperScripsit's complexity. This isn't the ultimate word processor, but a valid alternative to Scripsit.■

WordPerfect 4.0

★★★★★

WordPerfect 4.0 runs on the Tandy 1000, 1200, and 2000 (256K), requires two disk drives and MS-DOS 2.X. Satellite Software International, 288 W. Center St., Orem, UT 84057. 801-224-4000. $495 (includes mail-merge and spelling checker with 100,000-word dictionary).

I described Microsoft Word 2.0 as a "first-strike thermonuclear word processor" (August 1985, p. 114). However, I forgot that superpowers come in twos. WordPerfect 4.0, like Word, is an awesome program built for high-volume professional writing that is wasted on occasional correspondence. In many ways, it's even mightier than its Microsoft rival.

Most of WordPerfect's advantages involve extra convenience. It's not copy-protected (which I appreciate after seeing my one legal copy of Word disappear in a hard disk crash) and it can automatically save your file at specified intervals. The spelling dictionary is bigger. It can not only format columns of text but also add columns of numbers. And the screen display shows the page and line position indicator that Word inexplicably forgot.

But, WordPerfect isn't as dazzling in the "what you see is what you get" department: There's no on-screen justification or multiple windows, and less virtuosity at mixing dozens of fonts for a laser typesetter (though you can install up to five printers instead of the usual one). And it doesn't have an undelete function.

Compared to Word's layered alphabetic menus, WordPerfect's 40-plus commands (all done with the function and control, alternate, and shift keys) take extra memorization. The manual, while first-rate, is useless without the supplied function-key template.

With the color-coded template before you, you'll fly through mountainous papers or reports. Some programs can't print footnotes; WordPerfect automatically numbers and formats notes up to 16,000 lines long, not to mention doing indexes, tables of contents, and ThinkTank-style outlines. Some auxiliary programs such as SuperKey allow multikeystroke macros and file access passwords; WordPerfect has them built in.

Once you turn off its automatic hyphenation (it brings winged thoughts to a screeching halt a dozen times per page), WordPerfect will quickly and unobtrusively do any word processing job. Microsoft Word is flashier (on-screen boldface italics edited with a mouse), but WordPerfect is an unbeatable powerhouse. It's expensive, but definitive.

—Eric Grevstad

Telecommuter

★★★★★

Telecommuter runs on the Tandy 1000, 1200, and 2000 (256K) and requires one disk drive and MS-DOS 2.X. Sigea Systems, Inc., 19 Pelham Road, Weston, MA 02193. 617-647-1098. Write-It $125. XModem $200. Standard $200. Deluxe $300. Plus $400.

Telecommuter is an enhanced version of a program called Remote Control, which *80 Micro* reviewed in June 1985 (p. 113). As with Remote Control, Telecommuter provides a direct link between the Model 100/200 and a remote Tandy 1000/1200/2000 (which needs an auto-answer modem). You can access your PC over the phone to execute file transfers, DOS commands, and print documents, and even run programs. It is a significant enhancement for those who travel or use a portable when away from their PCs.

The different versions of Telecommuter are built around the same core program. Write-It only provides word-processing and fast file transfers. XModem includes protocol file transfer with the TELCOM mode. The Standard Telecommuter includes TELCOM and a host mode, and Deluxe provides access to the DOS and a multiple access level host mode. Telecommuter Plus has all the features of the other versions in one package.

The TELCOM mode is similar to the Model 100's, and there is a fast file transfer mode. The text processing mode uses many of the same commands as the 100/200's Text.

Telecommuter is better than the Remote Control program: The null modem cable is now sturdier and longer; there is single key redial in TELCOM mode from the PC; you have the option to automatically run application programs upon log-on in host mode; and there is a simulated sign-off if you lose your connection.

Also, text processing is more versatile. You can now append files to existing ones, or take them from disk and place them in text. You can divide large jobs into a series of small ones by using a command file to call files to be printed. You can write and print form letters. You can send printer output to the screen for preview or to a disk file.

Telecommuter can automatically sense whether you have a monochrome or color graphics board, but there is only one choice of display colors.

What was a very good manual is now even better. It has been split into two books, one for setting up and word processing, the other for telecommunications. The documentation leads you through the system, with many examples. Also enclosed are two reference cards with the communications and word processing commands.

Telecommuter links your 100/200 and your PC, giving you access to the PC's power while retaining your lap-top's portability.

—Thomas L. Quindry

How to Use Your Radio Shack Printer

★★★½

By William Bardin Jr. 204 pp. Softcover. Tandy/Radio Shack, One Tandy Center, Fort Worth, TX 76108. Radio Shack Catalog #26-1242. $14.95.

If you use any of the Radio Shack printers, whether it's a dot-matrix, daisy-wheel, or printer-plotter, then you need *How to Use Your Radio Shack Printer*. This book has an enormous amount of information, which at times is overwhelming. While it isn't thorough enough in some areas, no other source is as helpful for Radio Shack users.

This book covers all the printers carried by Radio Shack at the time it was printed: the CGP-115 and 220; the DMP-100, 110, 120, 200, 400, 420, 500, 2100, and 2100P; the DW I, II, and IIB; the DWP-210 and 410; the LP I, II, III, IV, V, VI, VII, and VIII; the QP I and II; the TP-10; and the Plotter/printer.

The later printers, such as the DMP-105, aren't included, but Barden notes that the newer printers can emulate at least one of the printers in the book. Even if your printer isn't listed, you can still use the book.

The book contains 12 chapters organized into three sections: Printer Basics, Printing Text, and Printing Graphics.

Printer Basics takes a brief look at the Radio Shack printer line, how printers form characters and communicate with computers, characters printed, simple programs for underlining and graphics, and a master index on the abilities of the various printers.

The next three chapters deal with printing text, first with normal text and simple word processing, then word processing functions such as wordwrap, justification, and proportional spacing. The last chapter in this section deals with such uses as mail labels, boilerplate form letters, and screen-printing text to your printer.

The final section tackles graphics: normal, screen, and creative printing. Normal printing uses the printer's built-in graphics characters to make boxes, graph forms, butterflies, and large characters.

The chapter on creative graphics shows you how to design characters and create pictures with direct dot-addressing. There's even a short section on using daisy-wheel printers to make graphs using the period and other characters.

Barden's book is well written, with many examples and dozens of printer hints. The hints are placed into sidebars, and give information about such things as the impression level and ribbon feed in daisy-wheels, or generating Japanese Kana symbols with the LP VIII and DMP-200, 400, 420, and 500.

The book's major fault is that it attempts to cover everything, while not providing enough in-depth information about any one printer. You need your printer manual and this book side-by-side.

One other limitation is that there aren't enough examples. This is especially true in the discussions on graphics.

Despite its problems, this is one book you should have if you own a Radio Shack printer or want to write programs that use standard Radio Shack printers.

—Terry Kepner

PRO-X-FTS

★★★★½

PRO-X-FTS runs on the Model 4/4P (64K), and requires one disk drive and an RS-232. Misosys Inc., P.O. Box 239, Sterling, VA 22170-0239. 703-450-4181. $24.95.

PRO-X-FTS is an XModem file transfer utility for making error-free transmissions between computers. It's not a full-featured telecommunications program. Instead, it's meant to be used along with a program such as COMM, which is supplied with TRSDOS 6.X.X.

XModem, the Ward Christensen protocol for error-free file transfer, is a de facto standard, and you can use it to download thousands of public domain programs.

If you use TRSDOS 6.2, you execute PRO-X-FTS from within your communications program by pressing clear/shift/0. With other DOSes (6.0, 6.1, DOS-PLUS IV), you must exit your communications program, invoke PRO-X-FTS, and return to the program once the file transfer is complete.

I used the program on a Model 4 running TRSDOS 6.2 to transfer a few programs from my Compaq, and it worked well.

The PRO-X-FTS utility is well worth the price, and makes error-free transfers easily, either locally between computers or from bulletin boards. I always wondered why the authors of TRSDOS and LDOS omitted XModem from COMM. Without it, LCOMM and COMM are only half the communication programs they could be. PRO-X-FTS makes them what they should be: useful.

—Gary Shade

LET THE SOURCE BE WITH YOU

TANDY 1000 W/2 DRIVES **$875.00**

Tandy 1000 w/1 Drive & 10 meg Hard Drive	$1439.00
Model 4D w/2 Double Sided Drives	$ 899.00
Tandy 1200 HD	$1549.00
IBM PCXT w/256K, and Two D5 Drives	$1795.00
IBM PCXT w/256K, One DS Drive & 10 meg Hard Disk	$2495.00
Color Card for IBM, TAND 1200 or Any IBM Clone	$ 130.00
128K Upgrade for Model 4D Installed	$ 50.00

DISK DRIVES

ONE YEAR WARRANTY ON TEAC

FD55B 40/40 trk DSDD bare	$ 99.00
FD55F 80/80 trk DSDD bare	$115.00
Tandon 40/40 trk bare	$115.00
8" Dual slimline case & power	$ 95.00!

ONE YEAR WARRANTY ON TEAC

FD55B 40/40 trk w/case & power supply	$144.00
FD55F 80/80 trk w/case & power supply	$160.00
Fullsize case & power supply	$ 45.00
Slimline case & power supply	$ 45.00
Dual slimline case & power supply	$ 55.00

PRINTERS

EPSON

LX80	$239.00
FX85+	$375.00
FX185+	$549.00
RX100	$399.00
LQ1500 & interface	$995.00
JX80	$595.00

STAR

SG10	$239.00
SG15	$379.00
SD10	$375.00
SD15	$475.00
SR10	$560.00
SR15	$640.00

DAISY WHEELS

Sanyo PR5000	$ 345.00
Epson DX10	$ 275.00
Star Powertype	$ 315.00
DWP220	$ 449.00
DWP510	$1075.00

Radio Shack Printers	Call for price
Radio Shack, IBM, TI, & Sanyo printer cables	$21.99
Printer paper 20# 2700 sheets	$25.00
Epson 80 series ribbon	$8.00
Epson 100 series ribbon	$10.00

PERIPHERALS

MODEMS

ANCHOR EXPRESS	
300/1200 baud (Hayes compatible)	$249.00
ANCHOR MARK X	
300 baud (Hayes compatible)	$115.00
Volksmodem 12 & cable	
300/1200 baud auto/ans au/dl	$205.00
Hayes 300 baud	$159.00
Hayes 1200 baud	$395.00
Volks modems & cable	$ 69.00

MONITORS

Teknika hi-res. color monitor	$310.00!!
Comrex green or amber monitor	$ 95.00
Zenith green or amber monitor	$ 95.00

MISCELLANEOUS

256K chips 150NS	$ 4.50
64K 150NS memory chips	$ .99
Power strip w/surge protection	$25.00
Bulk diskettes pack of 10	$10.00
Verbatim diskettes pack of 10	$20.00

Visit our two retail locations at:

886 Ecorse Road
Ypsilanti, MI 48197
(313) 426-5086 / (313) 482-4424
or
111 Marshall Street
Litchfield, MI 49252
(517) 542-3280
(517) 542-3939
(517) 542-3947

IMMEDIATE DELIVERY
DEALER INQUIRIES INVITED

TO ORDER: Call (313) 426-5086 or (313) 482-4424 or (517) 542-3280
(517) 542-3939 or (517) 542-3947

OR WRITE: DISPLAYED VIDEO, 111 MARSHALL ST., LITCHFIELD, MI 49252
OR 886 ECORSE RD., YPSILANTI, MI 48197

***TRS-80 is a trademark of the Tandy Corporation**
***IBM is a trademark of International Business Machines**
Prices & Specifications subject to change without notice

NEW FROM 80 MICRO

Classifieds

GET THE ATTENTION YOU DESERVE

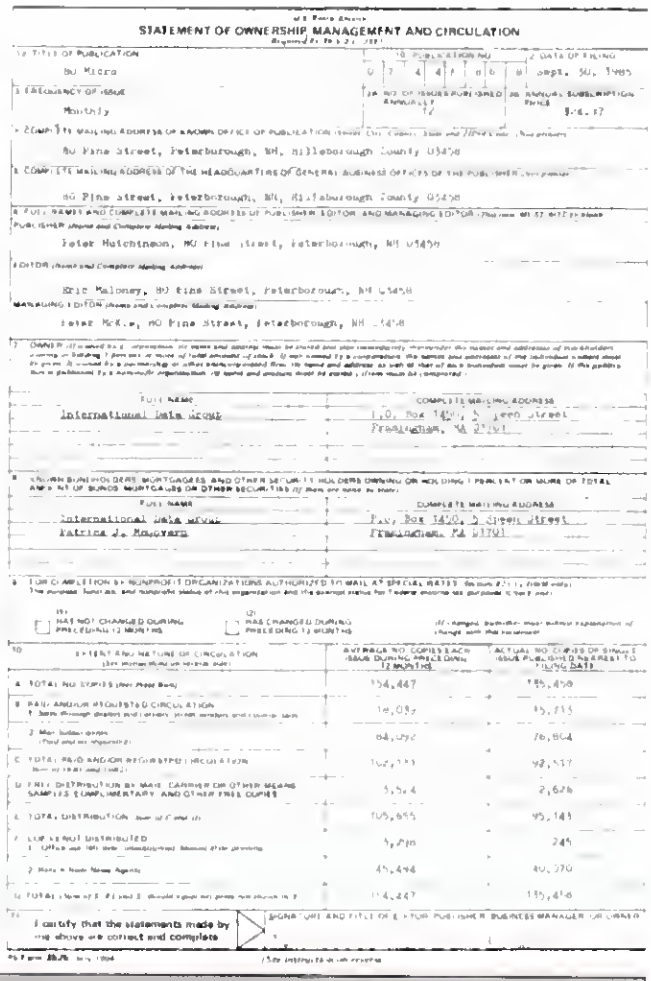
Tell more than 200,000 dedicated, interested TRS-80 users about your product or service with an efficient and economical ***80 Micro*** classified ad.

You'll reach the most people in the market for the least amount of money!

With ***80 Micro***'s well-established audience of involved buyers, sellers, and swappers, your ad is bound to get fast results!

For more information, write to:

80 Micro
Attn. Classified Manager
80 Pine Street
Peterborough, NH 03458



STATEMENT OF OWNERSHIP MANAGEMENT AND CIRCULATION

1a. TITLE OF PUBLICATION: 80 Micro — 1b. PUBLICATION NO.: 0744470 — 2. DATE OF FILING: Sept. 30, 1985

3. FREQUENCY OF ISSUE: Monthly — 3a. NO. OF ISSUES PUBLISHED ANNUALLY: 12 — 3b. ANNUAL SUBSCRIPTION PRICE: $24.00

4. COMPLETE MAILING ADDRESS OF KNOWN OFFICE OF PUBLICATION: 80 Pine Street, Peterborough, NH, Hillsborough County 03458

5. COMPLETE MAILING ADDRESS OF THE HEADQUARTERS OF GENERAL BUSINESS OFFICES OF THE PUBLISHER: 80 Pine Street, Peterborough, NH, Hillsborough County 03458

6. FULL NAMES AND COMPLETE MAILING ADDRESS OF PUBLISHER, EDITOR, AND MANAGING EDITOR
Publisher: Peter Hutchinson, 80 Pine Street, Peterborough, NH 03458
Editor: Eric Maloney, 80 Pine Street, Peterborough, NH 03458
Managing Editor: Peter McKay, 80 Pine Street, Peterborough, NH 03458

7. OWNER

Full Name	Complete Mailing Address
International Data Group	P.O. Box 1405, 5 Speen Street, Framingham, MA 01701

8. KNOWN BONDHOLDERS, MORTGAGEES, AND OTHER SECURITY HOLDERS OWNING OR HOLDING 1 PERCENT OR MORE OF TOTAL AMOUNT OF BONDS, MORTGAGES OR OTHER SECURITIES

Full Name	Complete Mailing Address
International Data Group	Box 1405, 5 Speen Street, Framingham, MA 01701
Patricia J. Mahoney	Framingham, MA 01701

9. FOR COMPLETION BY NONPROFIT ORGANIZATIONS AUTHORIZED TO MAIL AT SPECIAL RATES (Section 423.12, DMM only)
The purpose, function, and nonprofit status of this organization and the exempt status for Federal income tax purposes: (Check one)
(1) HAS NOT CHANGED DURING PRECEDING 12 MONTHS
(2) HAS CHANGED DURING PRECEDING 12 MONTHS (If changed, publisher must submit explanation of change with this statement)

10. EXTENT AND NATURE OF CIRCULATION	AVERAGE NO. COPIES EACH ISSUE DURING PRECEDING 12 MONTHS	ACTUAL NO. COPIES OF SINGLE ISSUE PUBLISHED NEAREST TO FILING DATE
A. TOTAL NO. COPIES (Net Press Run)	154,447	155,450
B. PAID AND/OR REQUESTED CIRCULATION 1. Sales through dealers and carriers, street vendors and counter sales	16,082	15,715
2. Mail Subscription (Paid and/or requested)	84,092	76,864
C. TOTAL PAID AND/OR REQUESTED CIRCULATION	100,174	92,579
D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES	5,504	2,626
E. TOTAL DISTRIBUTION (Sum of C and D)	105,678	95,141
F. COPIES NOT DISTRIBUTED 1. Office use, left over, unaccounted, spoiled after printing	4,769	245
2. Return from News Agents	44,444	60,270
G. TOTAL (Sum of E, F1 and 2 - should equal net press run shown in A)	154,447	155,456

11. I certify that the statements made by me above are correct and complete. — SIGNATURE AND TITLE OF EDITOR, PUBLISHER, BUSINESS MANAGER, OR OWNER

PS Form 3526, July 1984 (See instructions on reverse)

80 MICRO'S LIST of ADVERTISERS

Reader Service Number		Page
475	A-1 Computer Paper Company	129
82	Aerocomp	122, 123
215	Alcor Systems	51
136	Allen Gelder Software	148
30	Allwrite	10
476	AlphaBit Communications, Inc.	26
17	Alpha Products	13, 15
276	Alpine Data Services	121
374	ALPS	57
141	Anitek Software Products	23
319	Applied Creative Technologies	CII
383	Astro-Star Enterprises	111
152	BCCOMPCO	99
300	Bi-Tech Enterprises Inc.	105
315	Bible Research	152
186	Blue Ridge Software	147
328	Compco	129
86	Compulogic Corp	147
133	Computer Discount of America	118
357	Computer Friends	111
326	Computer Options Radio Shack	139
18	Computer Plus	141
532	Contract Services Associates	93
45	Comucopia Software	1
239	D & A Research	129
539	Desert Sound Inc.	63
282	DFW Computer Center	46, 47
536	DHA Systems	119
204	DiskCount Data	7
62	Displayed Video	127
91	Dotwriter	9
452	Dresselhaus Computer	119
216	E.A.P. Co.	72
85	Educational Micro Systems	75
	80 Micro	
	Back Issues	115
	Christmas	80
	Classified	126
549	Classified Ads	151
	Dealer Sell	99
	Foreign Dealers	89
	Load 80 Subscription	64, 65
	Subscription	117
	Subscription Problems	115
	Toll Free	116
	University Micro	152

Reader Service Number		Page
45	Electric Webster	1
134	Elek-Tek	101
161	Envision Design	99
178	EZWare Corp	62
214	Ft. Worth Computers	60, 61
295	GT Enterprises	111
185	Gooth Software	121
9	H & E Computronics	CIV
9	H & E Computronics	67
9	H & E Computronics	109
455	Hard Drive Specialist	145
355	H.D.P.	89
175	Howe Software	35
46	Hypersoft	79
373	Inmac	107
109	Instant Software	143
401	Intellitech Corp. (ITC)	42
101	J & M Systems, Ltd	CIII
126	JMG Software International	70
126	JMG Software International	2, 3
534	Jameco Electronics	79
518	James Halstead & Assoc	129
331	K Soft	104
*	Langley-St. Clair	63
339	Logical Systems	32
250	Marymac Industries Inc	105
363	MiCom	95
464	Micro Labs Inc	149
488	Micro-Link	104
514	Micro Mainframe	140
137	Miller Microcomputer Services	81
411	Montezuma Micro	36, 37
416	Montezuma Micro	90, 91
424	Montezuma Micro	77
281	Nibble Notch	133
232	Nocona Electronics	103
352	Northeast Peripherals	100
95	Northwest Computer Algorithms	100
127	Noteworthy Software, Inc	57
36	Omnisoft Research	83
151	Orion Instruments	105
470	P.G. Design Electronics	119
207	Pacific Exchanges	115, 129
200	Pacific Software Consultants	55
324	Pel/Tek	140

Reader Service Number		Page
124	Perry Computers	20
176	Personal Computer Products	137
440	Personal Integrated Computers	27
290	Pickles & Trout	56, 129
432	Pivar Computing Services Inc.	129
437	Portable Software	100
108	Powersoft	18, 19
308	Powersoft	4
249	Press A Software	136
76	Producer, The	24
174	Professional Tax Software	137
449	Professor Jones/Frogg House	150
30	Prosoft	11
512	Rockware Data Corp.	50
56	Ross Custom Electronics	148
203	Scientific Engineering Labs	129
371	Seatronics	150
503	Severts-Zorman Eng.	135
272	Simply The Best Software	129
430	Softech Information System	133
*	Software Support	96, 97
427	SOTA Computing Systems Ltd	78
525	Spectrum Holobyte	28
386	Statewide Microelectronics/ Digital Disks	85
71	Sublogic Communications	121
150	Summit Software Technology Inc.	89
285	Sun Research, Inc.	120
456	Sunlock Systems	62
266	T/Maker	73
472	T.N.T. Software	139
211	T Soft	82
189	Tab Sales	136
347	Talley Communications	79
81	Total Access	71
227	Trisoft	135
530	Zedcor	45

For further information from our advertisers, please use the Reader Service card.
***This advertiser prefers to be contacted directly**

Advertising Sales (603) 924-7138
or (800) 441-4403
West Coast Sales (415) 328-3470

Circle 432 on Reader Service card

Conversion Services

Any 9 track 1600 BPI MAGNETIC TAPE converted to:

Tandy Models I, III, 4/1000, 1200, 2000, II/12/16/6000, MS/dos, TRSdos, Xenix

8" CP/M	OS/6	WANG
IBM SYS/??	Display WR	MICOM
Macintosh	APPLE	ZENITH
IBM PC	TELEVIDEO	KAYPRO
MORROW	NORTHSTAR	XEROX
ALTOS		Many others

Disk to disk and disk to tape conversions for over 300 formats available.

Pivar Computing Services, Inc.
47 W. Dundee Rd.
Wheeling, IL 60090
(312)459-6010

Circle 516 on Reader Service card

REFER

- SAVES YOU HOURS PROGRAMMING.
- PROCESSES ANY LANGUAGE (ASM, BASIC, C, COBOL, PASCAL, ETC).
- PRETTY PRINTS PROGRAM SOURCE.
- CROSS-REFERENCES CONCURRENTLY: VARIABLES, VALUES, KEYWORDS, LINE NUMBERS AND/OR LABELS.
- FAST AND FLEXIBLE UTILITY
- SOURCE CODE & EXAMPLES INCL. SPECIFY MSDOS, TRSDOS, OR CPM

ONLY $39.

JAMES HALSTEAD & ASC.
1551 PLAINFIELD,
JOLIET, IL 60435
(815) 725-0346

Circle 475 on Reader Service card

PAPER

BUY WAREHOUSE DIRECT AND SAVE UP TO 40%

- **FANFOLD PAPER - 70+ TYPES IN STOCK** Blank, greenbar, smooth-edge, carbonless
- **LABELS AND LETTERHEADS**
- **37 WAREHOUSES NATIONWIDE**
- **NEXT DAY SHIPPING BY UPS** on most stock items - within 3 days for all

SAMPLE PRICES

9½x11 - 15 LB BLANK - 3300/CTN	27.95
14⅞x11 - 18 LB GREENBAR - 2800/CTN	34.95
9½x11 - 20 LB SMOOTH-EDGE - 2500/CTN	28.95

Call for shipping charges, cash and credit card discounts.

CALL TOLL FREE: **(800) 628-8736**
Open M-F 7-11AM & 1-5PM PST In CA, call (213) 804-1270

A-1 COMPUTER PAPER CO.
405 E. Third #206, Long Beach CA 90802
SEND CARD OR WRITE FOR FREE SAMPLES

Circle 203 on Reader Service card

ONE GPIB-488 INTERFACE

FOR ALL
IBM PC, XT, AT,
CLONES, APPLE MacINTOSH,
TANDY 2000, 1200HD, 1000

ANY LANGUAGE
EASY TO USE

MODEL 488-2000 PRICE $675
+ SHIPPING, INSURANCE & TAX
When ordering specify computer for proper cable.

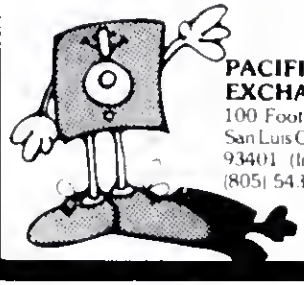
Scientific Engineering Laboratories
11 Neil Drive • Old Bethpage, NY 11804
Telephone: (516) 694-3370

Circle 207 on Reader Service card

wabash®

When it comes to Flexible Disks, nobody does it better than Wabash.

MasterCard. Visa Accepted.
Call Free: (800) 235-4137



PACIFIC EXCHANGES
100 Foothill Blvd
San Luis Obispo, CA
93401 (In Cal. call
(805) 543-1037)

Circle 239 on Reader Service card

DIGITAL and ANALOG I/O PORT KITS

Parallel 8 bit Input & Output for your Models I, III, 4 & CoCo

- Modular design for ADDITION of multiple ports
- LATCHED OUTPUT
- STATUS MONITORED BY LEDS
- SWITCH SELECTABLE I/O ADDRESS
- SOFTWARE DIAGNOSTICS

Complete I/O port kit (J107K)	**$35**
A-D/D-A Interface (J202K)	**$35**
CoCo Adapter—required for CoCo (J110K)	**$15**
Model III, 4 Adapter (J112K)	**$20**
5 Volt Power Supply (D100K)	**$25**
Complete Enclosure Kit (D100E)	**$30**
Relay Array (J207K-8)	**$32**

One **FREE POWER SUPPLY** with every four I/O Port Kits Ordered!
Add 30% for assembly.
Send check, money order, or C.O.D.

D & A RESEARCH
400 Wilson Avenue
Satellite Beach, FL 32937
305/777-1728

Circle 328 on Reader Service card.

ATTENTION MIII Owners

Fix Your MIII Computer for Only $50!*

Does Your MIII Have The Blues? Does it intermittenly REBOOT, LOSE DATA, or CRASH for no reason? The problem is probably the Mother Board. Replace this board YOURSELF for the cost of a service call! Anyone can do it Using Our Instructions.

For More Info and a FREE MIII Trouble-Shooting guide. Send a SASE to-

CompCo
Mountain Mall D-4
Gatlinburg Tn. 37738
Or Call 615-436-5189

*Exchange Price does not include shipping

Circle 290 on Reader Service card

CP/M FOR MOD II, 12, 16, 6000

P&T CP/M 2 is easy to use, flexible, & reliable — and backed by more than 5 years of experience. Contact:

PICKLES & TROUT®
P.O. Box 1206 • Goleta, CA 93116
(805) 685-4641

CP/M® DIGITAL RESEARCH

Circle 272 on Reader Service card.

SMALL C. COMPILER

Version 2.1

For Your Model II, III, IV running TRSDOS ©
PRICE: **$59.95**, including Assembler & Linking Loader

PLEASE SPECIFY VERSION

SIMPLY THE BEST SOFTWARE, INC.
2709 North Sibley Street
Metairie, LA 70003

©COPYRIGHT TANDY CORP

Listing 1 continued from p. 42

```
850 RETURN
860 ' ------------------------------------------------------------------
870 ' This routine handles a putchar statement. On entry, fpos will point
880 ' to the left paren of the function call.
890 ' ------------------------------------------------------------------
900 WHILE CPROG$(FPOS)<>"("; FPOS=FPOS+1: WEND
910 CK% = FPOS: ' save fpos
920 GOSUB 1820: ' get the parameter
930 IF TOKEN.TYP <> 4 THEN 900
940 FPOS = CK%
950 WHILE (CPROG$(FPOS) <> CHR$(39)): FPOS = FPOS+1: WEND: FPOS = FPOS + 1
960 IF CPROG$(FPOS) = "\" THEN FPOS = FPOS + 1: IF CPROG$(FPOS)="n" THEN CPROG$(
FPOS) = CHR$(13) ELSE IF CPROG$(FPOS) = "t" THEN CPROG$(FPOS) = CHR$(9) ELSE PRI
NT "Error - Illegal control character":STOP
970 PRINT CPROG$(FPOS);: FPOS = FPOS + 1: GOTO 1010
980 GOSUB 1820:CL%=TEMPVAR.COUNT+GLOBAL.COUNT:WHILE VAR.NAME$(CL%)<>TOKEN.VAL$AN
D CL%>=1 : CL%=CL%-1: WEND: 'find var
990 IF CL%=0 THEN PRINT"Putchar - Identifier not declared":STOP
1000 PRINT CHR$(VAR.INT%(CL%));
1010 WHILE (CPROG$(FPOS)<> ")"): FPOS = FPOS + 1: WEND
1020 RETURN
1030 ' ------------------------------------------------------------------
1040 ' This routine handles a getchar statement. On entry, fpos points to the
1050 ' character following the keyword getchar. The keyboard entry will be
1060 ' placed into the interpreter global function return variable, Func.ret.
1070 ' At exit, fpos will point to the character following the close paren of
1080 ' the function call.
1090 ' ------------------------------------------------------------------
1100 B$=INKEY$: IF B$="" THEN 1100
1110 FUNC.RET = ASC(B$)
1120 WHILE (CPROG$(FPOS)<>")"): FPOS = FPOS + 1: WEND
1130 RETURN
1140 ' ------------------------------------------------------------------
1150 ' Routine to handle the int declaration, during a function exec
1160 ' This routine merely places the name into the tempvar.name array, and
1170 ' sets the tempvar.val to zero.
1180 ' This routine assumes that the tempvar.name array has been initialized
1190 ' to hold null strings and tempvar.count was set to zero at init
1200 ' ------------------------------------------------------------
1210 WHILE TOKEN.VAL$<>";"
1220   GOSUB 1820: TEMPVAR.COUNT = TEMPVAR.COUNT + 1
1230   VAR.NAME$(GLOBAL.COUNT+TEMPVAR.COUNT) = TOKEN.VAL$
1240   VAR.INT%(GLOBAL.COUNT+TEMPVAR.COUNT) = 0
1250   GOSUB 1820
1260 WEND
1270 RETURN
1280 ' ------------------------------------------------------------------
1290 ' Routine to clear out the temporary variable arrays, and set local
1300 ' variable count to zero.
1310 ' ------------------------------------------------------------------
1320 FOR CL%= 1 TO TEMPVAR.COUNT
1330     VAR.NAME$(GLOBAL.COUNT+CL%) = "": VAR.INT%(GLOBAL.COUNT+CL%)=0
1340 NEXT CL%
1350 TEMPVAR.COUNT = 0
1360 RETURN
1370 ' ------------------------------------------------------------------
1380 ' This routine  gets called when the first token of a
1390 ' statement is not in one of the keywords recognized. In end, it
1400 ' should only be called when an identifier is located, as in an
1410 ' arithmetic statement. It will be assumed here that that is why
1420 ' this routine is being called.
1430 ' ------------------------------------------------------------------
1440 CL%=GLOBAL.COUNT+TEMPVAR.COUNT: B$=TOKEN.VAL$: WHILE (VAR.NAME$(CL%)<>B$) A
ND (CL% >=1): CL%=CL%-1: WEND
1450 IF CL%=0 THEN PRINT "Statement Error - Variable Not declared":STOP
1460 GOSUB 1820: ' get the next token
1470 IF TOKEN.VAL$="++" THEN VAR.INT%(CL%) = VAR.INT%(CL%)+1: RETURN
1480 IF TOKEN.VAL$="--" THEN VAR.INT%(CL%) = VAR.INT%(CL%)-1: RETURN
1490 IF TOKEN.VAL$<>"=" THEN PRINT "Syntax error": STOP
1500 GOSUB 1820
1510 IF TOKEN.TYP <>1 THEN 1660
1520 IF TOKEN.VAL$="GETCHAR" THEN GOSUB 1100: VAR.INT%(CL%)=FUNC.RET: RETURN
1530 CL1%=TEMPVAR.COUNT+GLOBAL.COUNT: WHILE VAR.NAME$(CL1%)<>TOKEN.VAL$ AND CL1%
>=1: CL1%=CL1%-1: WEND: IF CL1%=0 THEN PRINT "Variable Used - Not declared": STO
P
1540 GOSUB 1820: ' get token or operator
1550 IF TOKEN.TYP = 1 AND TOKEN.VAL$=";" THEN VAR.INT%(CL%)=VAR.INT%(CL1%): RETU
RN
1560 IF TOKEN.VAL$="++" THEN VAR.INT%(CL%)=VAR.INT%(CL1%)+1: RETURN
1570 IF TOKEN.VAL$="--" THEN VAR.INT%(CL%)=VAR.INT%(CL1%)-1: RETURN
1580 IF TOKEN.VAL$<>"+" THEN 1600
1590 GOSUB 1820:IF TOKEN.TYP=1 THEN CL2%=TEMPVAR.COUNT+GLOBAL.COUNT:WHILE VAR.NA
ME$(CL2%)<>TOKEN.VAL$:CL2%=CL2%-1:WEND:VAR.INT%(CL%)=VAR.INT%(CL1%)+VAR.INT%(CL2
%):RETURN ELSE VAR.INT%(CL%)=VAR.INT%(CL1%)+VAL(TOKEN.VAL$):RETURN
1600 IF TOKEN.VAL$<>"-" THEN 1620
1610 GOSUB 1820:IF TOKEN.TYP=1 THEN CL2%=TEMPVAR.COUNT+GLOBAL.COUNT:WHILE VAR.NA
ME$(CL2%)<>TOKEN.VAL$:CL2%=CL2%-1:WEND:VAR.INT%(CL%)=VAR.INT%(CL1%)-VAR.INT%(CL2
%):RETURN ELSE VAR.INT%(CL%)=VAR.INT%(CL1%)-VAL(TOKEN.VAL$):RETURN
1620 IF TOKEN.VAL$<>"*" THEN 1640
1630 GOSUB 1820:IF TOKEN.TYP=1 THEN CL2%=TEMPVAR.COUNT+GLOBAL.COUNT:WHILE VAR.NA
ME$(CL2%)<>TOKEN.VAL$: CL2%=CL2%-1:WEND:VAR.INT%(CL%)=VAR.INT%(CL1%)*VAR.INT%(CL
2%): RETURN ELSE VAR.INT%(CL%)=VAR.INT%(CL1%)*VAL(TOKEN.VAL$): RETURN
1640 IF TOKEN.VAL$<>"/" THEN PRINT"Assignment Statement Error":STOP
1650 GOSUB 1820:IF TOKEN.TYP=1 THEN CL2%=TEMPVAR.COUNT+GLOBAL.COUNT:WHILE VAR.NA
ME$(CL2%)<>TOKEN.VAL$:CL2%=CL2%-1:WEND:VAR.INT%(CL%)=VAR.INT%(CL1%)/VAR.INT%(CL2
%):RETURN ELSE VAR.INT%(CL%)=VAR.INT%(CL1%)/VAL(TOKEN.VAL$): RETURN
1660 VAR.INT%(CL%)=VAL(TOKEN.VAL$):HOLD.POS1 = FPOS:GOSUB 1820
1670 IF TOKEN.VAL$="," THEN FPOS=HOLD.POS1: RETURN
1680 CL1%=CL%: GOTO 1580
1690 ' ------------------------------------------------------------------
1700 ' This routine starts at fpos, and places the next input token
1710 ' (keyword, number, token) up to a delimiter into the string token$.
1720 ' Valid delimiters are : space, (, ), ;, [ , ], and comma and <cr>
1730 ' Note that upon exit, fpos is updated to point to the next character
1740 ' to be processed in the source program, following token$.
1750 ' Leading spaces and or tabs are ignored.
1760 ' The delimiter reached is returned in variable delim$.
1770 ' The Token.typ is either 1 = identifier, 2 = number(int), 3 = delim agn
1780 ' The token.val$ will have the string of the token found.
1790 ' token.typ of 4 = string in double quotes, 5 = string single quotes
1800 ' token.typ of 99 if fpos should get greater than eofc.
1810 ' ------------------------------------------------------------------
1820 DELIM$="":TOKEN.VAL$="":TOKEN.TYP=0
1830 WHILE (CPROG$(FPOS)=" ")OR(CPROG$(FPOS)=CHR$(9))OR(CPROG$(FPOS)=CHR$(13))
1840   FPOS = FPOS + 1
1850   IF FPOS >= EOFC THEN TOKEN.TYP=99: TOKEN.VAL$="": RETURN
1860 WEND
1870 IF FPOS = EOFC THEN TOKEN.TYP = 99:TOKEN.VAL$=CPROG$(FPOS): DELIM$=CPROG$(F
POS): RETURN
1880 DELIM$ = CPROG$(FPOS)
1890 IF CPROG$(FPOS+1)<>"" THEN 1940
1900 IF DELIM$<>"/" THEN 1940
1910 FPOS = FPOS + 2
1920 WHILE (CPROG$(FPOS)<>"/") : FPOS = FPOS+1: WEND
1930 IF (CPROG$(FPOS-1)<>"*") THEN FPOS = FPOS+1:GOTO 1920 ELSE FPOS=FPOS+1: GOT
O 1820
1940 IF DELIM$="{" OR DELIM$="}" OR DELIM$="(" OR DELIM$=")" OR DELIM$=" " OR DE
LIM$=CHR$(13) OR DELIM$=";" OR DELIM$="," THEN TOKEN.TYP = 3: TOKEN.VAL$=DELIM$:
 FPOS = FPOS + 1:RETURN
1950 IF DELIM$="=" THEN TOKEN.TYP=3: IF CPROG$(FPOS+1)="=" THEN TOKEN.VAL$="==":
FPOS=FPOS+2:RETURN ELSE TOKEN.VAL$=DELIM$:FPOS=FPOS+1:RETURN
1960 IF DELIM$="+" THEN IF CPROG$(FPOS+1)="+" THEN TOKEN.TYP=3:TOKEN.VAL$="++":
     FPOS = FPOS + 2: RETURN ELSE TOKEN.TYP=3: TOKEN.VAL$="+": FPOS=FPOS+1:
     RETURN
1970 IF DELIM$="-" THEN TOKEN.TYP=3: IF CPROG$(FPOS+1)="-" THEN TOKEN.VAL$ =
     "--": FPOS = FPOS+2: RETURN ELSE FPOS = FPOS+1: TOKEN.VAL$="-": RETURN
1980 IF DELIM$="<" THEN TOKEN.TYP=3: IF CPROG$(FPOS+1)="=" THEN TOKEN.VAL$="<=":
FPOS = FPOS+2:RETURN ELSE FPOS = FPOS + 1: TOKEN.VAL$="<": RETURN
1990 IF DELIM$="!" THEN TOKEN.TYP=3: IF CPROG$(FPOS+1)="=" THEN TOKEN.VAL$="!=":
FPOS = FPOS+2:RETURN ELSE FPOS=FPOS+1:TOKEN.VAL$="!": RETURN
2000 IF DELIM$="/" THEN TOKEN.TYP = 3: FPOS = FPOS + 1: TOKEN.VAL$="/": RETURN
2010 IF DELIM$="*" THEN TOKEN.TYP = 3: FPOS = FPOS + 1: TOKEN.VAL$="*": RETURN
2020 IF DELIM$ = CHR$(34) THEN  FPOS = FPOS + 1:TOKEN.VAL$="": WHILE (CPROG$(FPO
S)<>CHR$(34)): TOKEN.VAL$=TOKEN.VAL$+CPROG$(FPOS):FPOS = FPOS + 1: WEND : DELIM$
=CPROG$(FPOS): TOKEN.TYP = 4: FPOS = FPOS + 1:RETURN
2030 IF DELIM$=CHR$(39) THEN FPOS = FPOS + 1:TOKEN.VAL$="":WHILE (CPROG$(FPOS)<>
CHR$(39)): TOKEN.VAL$=TOKEN.VAL$+CPROG$(FPOS):FPOS=FPOS+1:WEND:DELIM$=CPROG$(FPO
```

Listing 1 continued

Listing 1 continued

```
5):TOKEN.TYP = 5:FPOS = FPOS+1:RETURN
2040 TOKEN.VAL$=CPROG$(FPOS): FPOS = FPOS + 1
2050 IDOK=0: WHILE (IDOK=0)
2060  IF CPROG$(FPOS)="!" THEN IDOK=1: GOTO 2160
2070 IF CPROG$(FPOS)="[" OR CPROG$(FPOS)="]" THEN IDOK=1:GOTO 2160
2080  IF CPROG$(FPOS)="(" OR CPROG$(FPOS)=")" THEN IDOK=1:GOTO 2160
2090  IF CPROG$(FPOS)=" " OR CPROG$(FPOS)=CHR$(13) THEN IDOK=1:GOTO 2160
2100  IF CPROG$(FPOS)="," OR CPROG$(FPOS)="," THEN IDOK=1: GOTO 2160
2110  IF CPROG$(FPOS)="+" OR CPROG$(FPOS)="-" THEN IDOK=1: GOTO 2160
2120  IF CPROG$(FPOS)="/" OR CPROG$(FPOS)="*" THEN IDOK=1: GOTO 2160
2130  IF CPROG$(FPOS)="<" OR CPROG$(FPOS)="="  THEN IDOK = 1: GOTO 2160
2140  TOKEN.VAL$=TOKEN.VAL$+CPROG$(FPOS)
2150  FPOS = FPOS + 1
2160 WEND
2170 DELIM$=LEFT$(TOKEN.VAL$,1)
2180 IF (DELIM$>="0" AND DELIM$<="9") THEN TOKEN.TYP = 2 ELSE TOKEN.TYP = 1
2190 DELIM$=CPROG$(FPOS)
2200 RETURN
2210 ' ---------------------------------------------------------------------
2220 ' This Routine Reads in an ASCII C program, generated from within
2230 ' BASIC, that was saved with the 'A' option. The text is loaded into
2240 ' the array CPROG$, which is to hold the entire C program that is to
2250 ' be interpreted. Upon return from this routine, fpos will be set to 1
2260 ' so that interpretation may begin.
2270 ' NOTE : This routine requires that the file name to be loaded appears
2280 '        in the string variable CFNAME$.
2290 '        The Array CPROG$(1500) must have been dimensioned at beginning
2300 ' ---------------------------------------------------------------------
2310 PRINT "Loading File : ";CFNAME$;"."
2320 OPEN "I",1,CFNAME$
2330 FPOS = 1
2340 WHILE NOT EOF(1)
2350   LINE INPUT# 1, CLINE$
2360  IF LEFT$(CLINE$,1)>="0" AND LEFT$(CLINE$,1)<="9" THEN CL%=INSTR(CLINE$," "
); CLINE$=RIGHT$(CLINE$,LEN(CLINE$)-CL%)
2370   FOR CL%=1 TO LEN(CLINE$)
2380       CPROG$(FPOS) = MID$(CLINE$,CL%,1)
2390       FPOS = FPOS + 1
2400   NEXT CL%
2410  CPROG$(FPOS) = CHR$(13): FPOS = FPOS + 1: ' Add a <cr> for looks if list
2420 WEND
2430 CLOSE 1
2440 PRINT "Loaded ";FPOS;" Characters."
2450 EOFC = FPOS: ' save off the total count of source characters
2460 FPOS = 1
2470 GLOBAL.COUNT = 0: TEMPVAR.COUNT = 0: FUNCTION.COUNT = 0
2480 PRINT "Performing String Garbage Collection...";:Z = FRE(A$):PRINT"Continui
ng"
2490 RETURN
2500 ' ---------------------------------------------------------------------
2510 ' This routine processes a FOR statement. Note that the FOR
2520 ' allowed here is severely limited. ONLY THE FORMAT BELOW WILL
2530 ' BE PERMITTED :
2540 '       for(varname = # or varname2; varname <= # or varname3; varname++)
2550 ' This is due to the great amount of code that would be needed for
2560 ' further enhancement of the powerful 'C' FOR statement.
2570 ' ---------------------------------------------------------------------
2580 GOSUB 1820: ' Get (
2590 GOSUB 1820: ' Get Varname for loop index
2600 CLF%=TEMPVAR.COUNT+GLOBAL.COUNT: WHILE (VAR.NAME$(CLF%)<>TOKEN.VAL$) AND (C
LF%>=1) :CLF%=CLF%-1: NEND: IF CLF%=0 THEN PRINT "Identifier: ";TOKEN.VAL$;" Not
 declared.";STOP
2610 IF CLF%=0 THEN PRINT "FOR - Index not declared":STOP
2620 XLF%=CLF%:INDEX.NAME$ = TOKEN.VAL$: 'save off index in var array
2630 GOSUB 1820:IF TOKEN.VAL$<>"=" THEN PRINT "FOR - Equal sign expected":STOP
2640 GOSUB 1820: IF TOKEN.TYP <>1 THEN 2670
2650 CLF%=TEMPVAR.COUNT+GLOBAL.COUNT:WHILE VAR.NAME$(CLF%)<>TOKEN.VAL$ AND CLF%>
=1: CLF%=CLF%-1:WEND: IF CLF%=0 THEN PRINT "For - Identifier not declared":STOP
2660 HOLD1F.VAL = VAR.INT%(CLF%):GOTO 2680
2670 HOLD1F.VAL = VAL(TOKEN.VAL$)
2680 GOSUB 1820: ' get ;
2690 GOSUB 1820: IF INDEX.NAME$<>TOKEN.VAL$ THEN PRINT"FOR - Must have index in
test part":STOP
2700 GOSUB 1820: IF TOKEN.VAL$<> "<=" THEN PRINT "FOR - only <= test allowed":ST
OP
2710 GOSUB 1820: IF TOKEN.TYP <> 1 THEN 2740
2720 CLF%=TEMPVAR.COUNT+GLOBAL.COUNT:WHILE VAR.NAME$(CLF%)<>TOKEN.VAL$ AND CLF%>
=1: CLF%=CLF%-1:WEND: IF CLF%=0 THEN PRINT"FOR - Limit variable not declared":ST
OP
2730 HOLD2F.VAL = VAR.INT%(CLF%): GOTO 2750
2740 HOLD2F.VAL = VAL(TOKEN.VAL$)
2750 GOSUB 1820: GOSUB 1820: IF TOKEN.VAL$<> INDEX.NAME$ THEN PRINT"For - increm
ent index only":STOP
2760 GOSUB 1820: IF TOKEN.VAL$<>"++" THEN PRINT"FOR - ++ expected":STOP
2770 GOSUB 1820: ' consume the end paren ')'
2780 GOSUB 1820: IF TOKEN.VAL$=";" THEN FOR XMF% = HOLD1F.VAL TO HOLD2F.VAL: VAR
.INT%(XLF%) = XMF%: NEXT XMF%: RETURN
2790 ' Must be a statement or a block of statements
2800 HOLDF.POS = FPOS-LEN(TOKEN.VAL$): ' save off current position in source for
 loop
2810 FOR XMF% = HOLD1F.VAL TO HOLD2F.VAL
2820   FPOS = HOLDF.POS: GOSUB 1820:IF TOKEN.VAL$ = "{" THEN HOLDF.EXIT$ = "}" E
LSE HOLDF.EXIT$ = ";" : ' reconsume first token, and set the exit token
2830   VAR.INT%(XLF%) = XMF%
2840   GOSUB 2920: ' process the statement or block
2850 NEXT XMF%
2860 RETURN  : ' done with the for loop !!!!
2870 ' ---------------------------------------------------------------------
2880 ' This routine processes a statement or a block of statements in
2890 ' a for loop, and then returns. It only processes the block one
2900 ' time per call.
2910 ' ---------------------------------------------------------------------
2920 WHILE TOKEN.VAL$<> HOLDF.EXIT$
2930   IF TOKEN.VAL$="PRINTF" THEN GOSUB 650  ELSE IF TOKEN.VAL$="PUTCHAR" THEN
GOSUB 900 ELSE IF TOKEN.VAL$="IF" THEN GOSUB 3140 ELSE IF TOKEN.VAL$="WHILE" THE
N GOSUB 3570 ELSE IF TOKEN.TYP=1 THEN GOSUB 1440
2940   GOSUB 1820 : ' get next token
2950 WEND
2960 IF TOKEN.VAL$ = "}" THEN TOKEN.VAL$="": ' fix up so dont quit interp.
2970 RETURN
2980 ' ---------------------------------------------------------------------
2990 ' This routine skips a block between braces or up to a ;, dependent
3000 ' upon hold.exit$. It is used in IF processing.
3010 ' ---------------------------------------------------------------------
3020 WHILE TOKEN.VAL$ <> HOLDI.EXIT$
3030   GOSUB 1820
3040 WEND
3050 IF TOKEN.VAL$ = "}" THEN TOKEN.VAL$="": ' fix up so dont quit
3060 RETURN
3070 ' ---------------------------------------------------------------------
3080 ' This routine processes an IF statement. Note that only the simplest
3090 ' form of an IF is allowed, that is, as below :
3100 '   IF (varname [=,<,>,!=] # [varname])  expr ELSE expr
3110 ' Blocks of statements may be in the if, but no FOR loops may exist here
3120 ' This is due to the non-recursiveness of BASIC
3130 ' ---------------------------------------------------------------------
3140 GOSUB 1820: ' get the (
3150 GOSUB 1820 : ' get the varname
3160 CLI%=TEMPVAR.COUNT+GLOBAL.COUNT:WHILE VAR.NAME$(CLI%)<>TOKEN.VAL$ AND CLI%>
=1 : CLI%=CLI%-1: WEND: IF CLI%=0 THEN PRINT "IF - Variable not declared";STOP E
LSE XLI%=CLI%
3170 GOSUB 1820: HOLDI.OPR$=TOKEN.VAL$: ' get operator and save it
3180 GOSUB 1820: IF TOKEN.TYP<>1 THEN 3210
3190 CLI%=TEMPVAR.COUNT+GLOBAL.COUNT:WHILE VAR.NAME$(CLI%)<>TOKEN.VAL$ AND CLI%>
=1 : CLI%=CLI%-1: WEND: IF CLI%=0 THEN PRINT"IF - Variable not declared";STOP
3200 HOLDI.VAL = VAR.INT%(CLI%) : GOTO 3220
3210 HOLDI.VAL = VAL(TOKEN.VAL$)
3220 GOSUB 1820: ' get closing paren )
3230 DO.ELSE = 0
3240 GOSUB 1820: IF TOKEN.VAL$<>"{" THEN HOLDI.EXIT$=";" ELSE HOLDI.EXIT$="}"
3250 IF HOLDI.OPR$ <> "==" THEN 3280
3260   IF VAR.INT%(XLI%) = HOLDI.VAL THEN GOSUB 3470: DO.ELSE = 1: GOTO 3370
3270   GOSUB 3820 : GOTO 3370
3280 IF HOLDI.OPR$ <> "<" THEN 3310
3290  IF VAR.INT%(XLI%) < HOLDI.VAL THEN GOSUB 3470: DO.ELSE = 1: GOTO 3370
3300  GOSUB 3820: GOTO 3370
3310 IF HOLDI.OPR$ <> ">" THEN 3340
3320  IF VAR.INT%(XLI%) > HOLDI.VAL THEN GOSUB 3470: DO.ELSE = 1: GOTO 3370
```

Listing 1 continued

Listing 1 continued

```
3330  GOSUB 3820: GOTO 3370
3340 IF HOLDI.OPR$ <> "|=" THEN PRINT "IF - Invalid compare operator":STOP
3350  IF VAR.INT%(XLI%) <> HOLDI.VAL THEN GOSUB 3470: DO.ELSE = 1: GOTO 3370
3360  GOSUB 3820: GOTO 3370
3370 HOLDI.POS2 = FPOS
3380 GOSUB 1820: ' get next token
3390 IF TOKEN.VAL$<>"ELSE" THEN FPOS = HOLDI.POS2: RETURN
3400 IF DO.ELSE=0 THEN GOSUB 1820:GOSUB 3470: RETURN
3410 GOSUB 1820: IF TOKEN.VAL$ <> "{" THEN HOLDI.EXIT$=";" ELSE HOLDI.EXIT$="}"
3420 GOSUB 3820: RETURN
3430 ' ------------------------------------------------------------------
3440 ' This routine handles an if block or statement that is either the
3450 ' valid if part or the else part.
3460 ' ------------------------------------------------------------------
3470 WHILE TOKEN.VAL$ <> HOLDI.EXIT$
3480  IF TOKEN.VAL$="PRINTP" THEN GOSUB 650 ELSE IF TOKEN.VAL$="PUTCHAR" THEN
     GOSUB 900 ELSE IF TOKEN.VAL$="FOR" THEN GOSUB 2500 ELSE IF TOKEN.VAL$="WHIL
E" THEN GOSUB 3570 ELSE IF TOKEN.TYP = 1 THEN GOSUB 1440
3490  GOSUB 1820 : ' get next token
3500 WEND
3510 IF TOKEN.VAL$="}" THEN TOKEN.VAL$="": ' fix up so do not quit yet
3520 RETURN
3530 ' ------------------------------------------------------------------
3540 ' This routine handles the while statement. Note that only the simple
3550 ' conditional operators are allowed. No Ands or Ors !
3560 ' ------------------------------------------------------------------
3570 GOSUB 1820: ' Get the (
3580 GOSUB 1820: ' Get the variable. Note, it must be a variable name
3590 CLW%=TEMPVAR.COUNT+GLOBAL.COUNT:
      WHILE (VAR.NAME$(CLW%)<> TOKEN.VAL$) AND (CLW%>=1) :
         CLW%=CLW%-1:
      WEND
3600 IF CLW%=0 THEN PRINT"While - Variable Not Declared":STOP
3610 XLW% = CLW%
3620 GOSUB 1820: HOLD.OPRW$=TOKEN.VAL$: ' Get operator, check it later
3630 GOSUB 1820: IF TOKEN.TYP <> 1 THEN 3660
3640 CLN%=TENPVAR.COUNT+GLOBAL.COUNT: WHILE VAR.NAME$(CLW%)<>TOKEN.VAL$ AND CLW%
>=0: CLW%=CLW%-1:WEND: IF CLW%=0 THEN PRINT"While - Variable.not declared":STOP
3650 HOLD.VALW = VAR.INT%(CLW%): GOTO 3670
3660 HOLD.VALW = VAL(TOKEN.VAL$)
3670 GOSUB 1820: ' get closing paren )
3680 GOSUB 1820: HOLD.POSW = FPOS - LEN(TOKEN.VAL$)
3690 IF HOLD.OPRW$ <> "<" THEN 3760
3700 WHILE (VAR.INT%(XLW%) < HOLD.VALW)
3710   FPOS = HOLD.POSW
3720   GOSUB 1820: IF TOKEN.VAL$="{" THEN HOLD.EXITW$="}" ELSE HOLD.EXITW$=";"
3730   GOSUB 4000: ' process a block or statement
3740 WEND
3750 RETURN
3760 IF HOLD.OPRW$ <> ">" THEN 3830
3770 WHILE (VAR.INT%(XLW%) > HOLD.VALW)
3780   FPOS = HOLD.POSW
3790   GOSUB 1820: IF TOKEN.VAL$="{" THEN HOLD.EXITW$="}" ELSE HOLD.EXITW$=";"
3800   GOSUB 4000: ' go process the statement
3810 WEND
3820 RETURN
3830 IF HOLD.OPRW$ <> "==" THEN 3900
3840 WHILE (VAR.INT%(XLW%) = HOLD.VALW)
3850   FPOS = HOLD.POSW
3860   GOSUB 1820: IF TOKEN.VAL$="{" THEN HOLD.EXITW$="}" ELSE HOLD.EXITW$=";"
3870   GOSUB 4000: ' go process statement
3880 WEND
3890 RETURN
3900 IF HOLD.OPRW$ <> "|=" THEN PRINT "While - Invalid Conditional Operator":STO
P
3910 WHILE (VAR.INT%(XLW%) <> HOLD.VALW)
3920   FPOS = HOLD.POSW
3930   GOSUB 1820: IF TOKEN.VAL$="{" THEN HOLD.EXITW$="}" ELSE HOLD.EXITW$=";"
3940   GOSUB 4000: ' go process statement
3950 WEND
3960 RETURN
3970 ' ------------------------------------------------------------------
3980 ' This routine handles the statement blocks for the while statement.
3990 ' ------------------------------------------------------------------
4000 WHILE TOKEN.VAL$ <> HOLD.EXITW$
4010  IF TOKEN.VAL$="PRINTF" THEN GOSUB 650 ELSE IF TOKEN.VAL$="PUTCHAR" THEN GO
SUB 900 ELSE IF TOKEN.VAL$="IF" THEN GOSUB 3140 ELSE IF TOKEN.VAL$="FOR" THEN GO
SUB 2500 ELSE IF TOKEN.TYP=1 THEN GOSUB 1440
4020  GOSUB 1820: ' get next token
4030 WEND
4040 IF TOKEN.VAL$="}" THEN TOKEN.VAL$="": ' fix up so as not to quit
4050 RETURN
```

End

Program Listing 2. Demo of PRINTF statement.

```
10 MAIN()
20 {
30    PRINTF("\nHello World\n");      /* PRINT A MESSAGE */
40 }
```

End

Program Listing 3. Demo of While statement. Copies input to output.

```
10 MAIN()
20 {
30   INT INT1;
40   PRINTF("\nCopy from input to output. Note this is slow\n");
50   PRINTF("Please press any keys, wait for display, CTRL-Q to quit\n");
60   WHILE (INT1 != 17)        /* LOOP UNTIL THE CTRL-Q IS PRESSED */
70    {   INT1 = GETCHAR();
80        PUTCHAR(INT1);
90    }
100  PRINTF("\nTest Completed\n");    /* TELL THEM WE ARE DONE */
110 }
```

End

Program Listing 4. Demo of For loop.

```
10 MAIN()
20 {
30    INT I1,I2,I3,I4;
40    PRINTF("\nExample of a For Loop\n");
50    I2 = 2 + I3;
60    I4 = 7;
70    FOR (I1=I2;I1<=I4;I1++)          /* LOOP FROM 2 TO 7 */
80    {
90       I2++;
100       PRINTF("In Loop, i1 = %d, i2 = %d\n",I1,I2);
110   }
120   PRINTF("\nOut of Loop Successfully\n");
130 }
```

End

Program Listing 5. Demo of If statement, the single statement type.

```
10 MAIN()
20 {
30     INT IN1, IN2, IN3, IN4;
40     PRINTF("\n\nDemonstration of the IF Statement\n");
50     IN1 = 100;
60     IN3 = 50 + IN1;
```

Listing 5 continued

Listing 5 continued

```
    70     PRINTF("IN1 = %d, and IN3 = %d\n",IN1,IN3);
    80     PRINTF("Since in3 > in1, you will see the greater message\n");
    90     IF (IN3 > IN1)
   100       PRINTF("\nIN3 is greater than IN1\n");
   110     ELSE
   120       PRINTF("\nIN3 is Not greater than IN1\n");
   130  }
```

End

Program Listing 6. Demo of For loop, used to print a table of squares for 1 to 10.

```
    10 MAIN()
    20 {
    30    INT I1,J1,K1;
    40    PRINTF("\n\nTable of squares\n");
    50    PRINTF("-----------------\n");
    60    FOR (I1=1;I1<=10;I1++)
    70    {   K1 = I1 * I1;
    80        PRINTF("\n%d , %d",J1,K1);
    90    }
   100    PRINTF("\n-----------------\n");
   110 }
```

End

Program Listing 7. Demo of If statement nested in a For loop to display odd and even numbers between 1 and 10.

```
    10 MAIN()
    20 {
    30    INT I1, I2, I3, I4;
    40    PRINTF("\nTable of odd and even\n");
    50    PRINTF("-----------------\n");
    60    FOR (I1=1;I1<=10;I1++)
    70    {   I2 = I1 / 2;
    80        I3 = I2 * 2;
    90        IF (I3 == I1)
   100            PRINTF("%d , even\n",I1);
   110        ELSE
   120            PRINTF("%d , odd\n",I1);
   130    }
   140    PRINTF("\n-----------------\n");
   150 }
```

End

Program Listing 8. Demo of While loop nested within For loop. Note that interpreter will not allow For or While loops to be nested, but will work with one of each.

```
    10 MAIN()
    20 {
    30    INT I1, I2, I3, I4, I5;
    40    PRINTF("\nDemonstration of Nested For and while loops\n");
    50    FOR (I1=1;I1<=4; I1++)
    60    {
    65        PRINTF("In For, I1 = %d\n",I1);
    70        I2 = 0;
    80        WHILE (I2 < 2)
    81        {
    90            PRINTF("In While, I2 = %d\n",I2);
    91            I2++;
    92        }
   100    }
   110    PRINTF("\n\nDemo finished\n");
   120 }
```

End

Listing continued from p. 57

```
1610 REM  ** Convert buffer variables to A( )  **
1620 GET 1,R:A(2)=CVI(QB$):A(3)=CVI(FN$):A(4)=CVI(FA$):A(5)=CVS(FP$)
1630 A(6)=CVI(FT$):A(7)=CVI(P2$):A(8)=CVS(F3$):A(9)=CVI(P$)
1640 A(10)=CVI(SA$):A(11)=CVI(GO$):A(12)=CVI(DR$):A(13)=CVI(TR$)
1650 A(14)=CVI(RA$):A(15)=CVI(TN$):A(16)=CVI(AA$):A(17)=CVI(DP$)
1660 A(18)=CVI(ST$):A(19)=CVI(BS$):A(20)=CVI(FO$):RETURN
1670 REM  ** Subroutine to print stats from PGM to Fouls **
1680 GOSUB 1220 :LPRINT USING T$; A(3);:GOSUB 1290 :LPRINT USING T$; A(4);
1690 GOSUB 1230 :LPRINT USING T$; A(5);:GOSUB 1220
1700 FOR Y= 6 TO 7:LPRINT USING T$; A(Y);:NEXT Y:GOSUB 1230
1710 LPRINT USING T$; A(8);:GOSUB 1290 :LPRINT USING T$; A(9);:GOSUB 1240
1720 LPRINT USING T$; A(10);:GOSUB 1220
1730 FOR Y= 11 TO 13:LPRINT USING T$; A(Y);:NEXT Y:GOSUB 1240
1740 LPRINT USING T$; A(14);:GOSUB 1220
1750 FOR Y=15 TO 20:LPRINT USING T$; A(Y);:NEXT Y:LPRINT" ":RETURN
1760 REM  ** Convert T(x,y) to A(y)  **
1770 FOR Y=1 TO 20:A(Y)=T(X,Y):NEXT Y:RETURN
1780 REM  ** Main Menu **
1790 CLS:PRINTTAB(21)"Basketball  Statistics"
1800 PRINT TAB(38)"Menu"
1810 PRINT TAB(6)"A) Initialize information for a new team"
1820 PRINT TAB(6)"B) Add a player to the roster"
1830 PRINT TAB(6)"C) Make corrections to previously entered information"
1840 PRINT TAB(6)"D) Type in statistics for a game"
1850 PRINT TAB(6)"E) Print the team record"
1860 PRINT TAB(6)"P) Print team totals (game by game - with opponents)
1870 PRINT TAB(6)"G) Print team totals (with up-to-date totals for players)"
1880 PRINT TAB(6)"H) Print statistics for an individual player"
1890 PRINT TAB(6)"I) Print statistics for a particular game"
1900 PRINT TAB(6)"J) Exit the program"
1910 PRINT:PRINT TAB(4)"Enter your choice (A - J) from above:";
1920 T$="":T=0
1930 POKE 16409,1:T$=INKEY$:IF LEN(T$)=0 THEN 1930
1940 T=INSTR("ABCDEFGHIJ",T$):IF T=0 THEN 1930
1950 POKE 16409,0
1960 ON T GOSUB 4590,4480,3390,1990,2510,2680,2730,2910,3120,120
1970 GOTO 1790
1980 REM  ** Type in statistics for a game **
1990 CLS:PRINT TAB(23)"Update Statistics";GOSUB 270:PRINT TAB(13) S$;"    ";Y$
2000 PRINT TAB(28)"Game #";G+1:PRINT TAB(3) STRING$(58,140):POKE 16916,4
2010 PRINT:PRINT"This section allows you to type in the statistics for"
2020 PRINT"a game. All files will be updated. Are you certain that"
2030 PRINT"you wish to type in game statistics?":GOSUB 210
2040 IF T$="N" THEN 2090
2050 G=G+1:GOSUB 440  :GOSUB 360
2060 PRINT@258, CHR$(31)"  Please type in the following information:"
2070 PRINT@389, "Name of opponent"; : INPUT O$
2080 IF LEN(O$)>10 THEN PRINT"Name can not exceed 14 letters.":GOSUB 160:
     GOTO 2060
2090 PRINT@517, CHR$(31);"Date of game (MM/DD/YY)";:INPUT D$
2100 IF LEN(D$)<>8 THEN 2090
2110 PRINT@645, "Home or Away";:INPUT H$:GOSUB 200  :IF T$="N" THEN 2060
2120 CLS:PRINT@258, "Now enter the statistics for each player on"
2130 PRINT"your roster. You will be asked if you wish to update"
2140 PRINT"each player. If you answer 'yes', then you will type"
2150 PRINT"in the numbers for each of the 13 categories. "
2160 PRINT"Remember that you may just press <ENTER> to put"
2170 PRINT"in a zero for any category."
2180 PRINT:PRINT"  When you have completed entering the statistics"
2190 PRINT"for a player, you will be asked if these are correct."
2200 PRINT"If you have made an error, you will need to re-enter all"
2210 PRINT"the statistics for that player.":GOSUB 160  :GOSUB 490
2220 REM  ** Players Input **
2230 FOR X=1 TO P:GOSUB 510
2240 PRINT@258, CHR$(31); "Do you wish to update "; P$(X);" (Y/N)?":GOSUB 210
2250 IF T$="N" THEN 2300
2260 T(X,1)=T(X,1)+1:A(1)=1
2270 PRINT@258,CHR$(31);P$(X);:PRINT@280,"# ";N$(X);:PRINT@269,"Games:"T(X,1)
2280 PRINT@385,"Quarters";:INPUT A(2):GOSUB 530  :GOSUB 200
2290 IF T$="N" THEN 2270
2300 GOSUB 660  :GOSUB 750  :GOSUB 770  :GOSUB 800  :GOSUB 880  :NEXT X
2310 REM  ** Team Totals **
2320 GOSUB 980:GOSUB 1060:X=21:GOSUB 750:GOSUB 770:GOSUB 800:GOSUB 880
2330 REM  ** Opponent's totals **
2340 X=22:GOSUB 510
2350 PRINT@258,CHR$(31);P$(X):A(1)=1:A(2)=4
2360 GOSUB 530  :GOSUB 200  :IF T$="N" THEN 2350
2370 IF A(4)>0 THEN A(5)=A(3)/A(4)*100
2380 IF A(7)>0 THEN A(8)=A(6)/A(7)*100
2390 A(9)=2*A(3)+A(6):ST=A(9):A(10)=A(9):A(13)=A(11)+A(12):A(14)=A(13)
2400 FOR Y1=1 TO 20:T(X,Y1)=T(X,Y1)+A(Y1):NEXT Y1
2410 IF T(X,4)>0 THEN T(X,5)=T(X,3)/T(X,4)*100
2420 IF T(X,7)>0 THEN T(X,8)=T(X,6)/T(X,7)*100
2430 IF G>0 THEN  T(X,18)=T(X,9)/T(X,1):T(X,14)=T(X,13)/T(X,1)
2440 GOSUB 750  :GOSUB 770  :GOSUB 800  :GOSUB 880
2450 REM  ** Determine Winner **
2460 IF SU>ST THEN W=W+1 ELSE L=L+1
2470 REM  ** Print to files **
2480 GOSUB 250  :GOSUB 310  :GOSUB 1020
2490 POKE 16916,0:RETURN
2500 REM ** Print Team Record **
2510 GOSUB 270:IF G=0 THEN CLS:PRINT"No games have been played.":GOSUB 160:
     RETURN
2520 GOSUB 1320 :GOSUB 1050 :GOSUB 1080 :LPRINT TAB(60) "      Score"
2530 LPRINT TAB(32) "Game   Date   Opponent          Us    Opponent"
2540 LPRINT TAB(32) "----- -------- --------------- ------- --------"
2550 T$=" ##  %      % %              %   ###     ###"
2560 OPEN"I",3, "GAMES/TXT"
2570 FOR X=1 TO G:INPUT#3,O$,D$,H$,SU,ST:LPRINT TAB(32) USING T$;X,D$,O$,SU,ST
2580 NEXT X:CLOSE:RETURN
2590 REM  ** Print team totals - with opponents  **
2600 GOSUB 270
2610 IF G=0 THEN CLS:PRINT"No games have been played":GOSUB 160  :RETURN
2620 GOSUB 1320 :GOSUB 1050 :GOSUB 1080 :LPRINT" ":GOSUB 1130 :GOSUB 1270
2630 LPRINT T$;:GOSUB 1150 :T$ = RIGHT$(T$,78):LPRINT T$:GOSUB 1280
2640 LPRINT T$:X=21:GOSUB 750  :GOSUB 770
2650 OPEN"I",3,"GAMES/TXT"
2660 FOR R=1 TO G:INPUT#3,O$,D$,H$,SU,ST:GOSUB 1210 :LPRINT USING T$;R;
2670 GOSUB 1250 :LPRINT USING T$;D$;:GOSUB 1260 :LPRINT USING T$;O$;
2680 LPRINT USING " %  %"; H$;:GOSUB 1620 :GOSUB 1680 :NEXT R:CLOSE
2690 GOSUB 360 :X=21:GOSUB 1770 :LPRINT" ":LPRINT STRING$(112,"-"):LPRINT " "
2700 LPRINT TAB(12)"Totals"; STRING$(13,32);:GOSUB 1680 :LPRINT " "
2710 LPRINT BTRING$(112,"-"):RETURN
2720 REM  ** Print team totals - with players  **
2730 GOSUB 270  :GOSUB 360  :GOSUB 440  :GOSUB 1320 :GOSUB 1050 :GOSUB 1080
2740 LPRINT TAB(47) USING"## Game Totals";G:LPRINT" ":LPRINT" "
2750 GOSUB 1130 :GOSUB 1160 :LPRINT " # Player"; TAB(26) T$:GOSUB 1170
2760 LPRINT T$;:GOSUB 1180 :LPRINT T$
2770 FOR X=1 TO P:GOSUB 1770 :GOSUB 2820 :NEXT X
2780 FOR X = 21 TO 22:LPRINT STRING$(112,"-"):LPRINT" ":GOSUB 1770
2790 GOSUB 2820 :IF G=0 THEN 2810
2800 GOSUB 2860 :GOSUB 2820
2810 LPRINT" ":NEXT X:LPRINT BTRING$(112,"-"):RETURN
2820 GOSUB 1180 :LPRINT USING T$; N$(X);:GOSUB 1200 :LPRINT USING T$; P$(X);
2830 LPRINT" ";:GOSUB 1210 :LPRINT USING T$; A(1);:GOSUB 1220
2840 LPRINT USING T$; A(2);:GOSUB 1680 :IF X/3=INT(X/3) AND X<20 THEN LPRINT" "
2850 RETURN
2860 A(1)=1:A(2)=4:FOR Y=3 TO 4:A(Y)=A(Y)/G:NEXT Y
2870 FOR Y=6 TO 7:A(Y)=A(Y)/G:NEXT Y:A(9)=A(9)/G
2880 FOR Y=11 TO 13:A(Y)=A(Y)/G:NEXT Y:FOR Y=15 TO 20:A(Y)=A(Y)/G : NEXT Y
2890 IF X=21 THEN P$(X)="Team - per game" ELSE P$(X)="Opp. - per game":RETURN
2900 REM  ** Print statistics for an individual player **
2910 GOSUB 270  :GOSUB 440  :CLS:T$ ="Print Statistics for an Individual Player"
2920 PRINT TAB(11) T$:T$=$$+"    "+Y$:T=INT((63-LEN(T$))/2)
2930 PRINT:PRINT TAB(T) T$:PRINT:PRINT TAB(28)"Roster":PRINT
2940 FOR X = 1 TO P STEP 2:PRINT X; P$(X);
2950 IF X=P THEN PRINT ELSE PRINT TAB(32) X+1; P$(X+1)
2960 NEXT X:PRINT:PRINT"Type the number of the player you want:";
2970 LINE INPUT T$:T1=VAL(T$)
2980 IF T1<0 OR T1>P THEN PRINT"That is not a choice.":GOSUB 160  :RETURN
2990 X=T1:GOSUB 750  :GOSUB 1320 :GOSUB 1050 :GOSUB 1080 :T$="Player: "+P$(X)
3000 GOSUB 1110 :LPRINT TAB(T) T$:T$="Jersey #"+N$(X):GOSUB 1110
3010 LPRINT TAB(T) T$:LPRINT" ":LPRINT" ":GOSUB 1130 :GOSUB 1160
3020 LPRINT"G#   Date   Opponent"; TAB(28) RIGHT$(T$,82):GOSUB 1180
3030 T$="-- -------- "+STRING$(15,"-")+RIGHT$(T$,84):LPRINT T$
3040 OPEN"I",3,"GAMES/TXT":GOSUB 770
3050 FOR R=1 TO G:GOSUB 1620 :INPUT#3,O$,D$,N$,SU,ST:GOSUB 1210
3060 LPRINT USING T$;R;:GOSUB 1250 :LPRINT USING T$;D$;
3070 LPRINT USING " %                %"; O$;:GOSUB 1220 :LPRINT USING T$;A(2);
3080 GOSUB 1680 :NEXT R:LPRINT STRING$(112,"-"):CLOSE:GOSUB 360
3090 X=T1:GOSUB 1770 :LPRINT TAB(12) "Totals";:LPRINT TAB(28) USING "###";A(2);
3100 GOSUB 1660 :LPRINT STRING$(112,"-"):RETURN
3110 REM  ** Print statistics for a perticular game **
3120 GOSUB 270  :GOSUB 440
```

Listing continued

Listing continued

```
3130 IF G=0 THEN CLS:PRINT"There are no games played.":GOSUB 160  :GOTO 470
3140 CLS:PRINT TAB(12) "Print Statistics for a Particular Game"
3150 PRINT TAB(29) "Games:":PRINT
3160 OPEN"I",3,"GAMES/TXT"
3170 FOR Y1=1 TO G:INPUT#3,O$,D$,H$,SU,ST:PRINT Y1;O$;" - ";H$,
3180 IF Y1<>20 THEN 3200
3190 FOR X1=1 TO 1E3:NEXT X1
3200 NEXT Y1:CLOSE:PRINT
3210 PRINT:PRINT"Choose the number of the game you would like to see.";
3220 LINE INPUT T$:T1=VAL(T$)
3230 IF T1<0 OR T1>G THEN PRINT"That is not one of the choices.":GOSUB 160:RETURN
3240 R=T1:GOSUB 1329 :GOSUB 1958 :OPEN"I",3,"GAMES/TXT"
3250 FOR X=1 TO R:INPUT#3,O$,D$,H$,SU,ST:NEXT X:CLOSE:GOSUB 1300
3260 LPRINT TAB(34) "Game #";LPRINT USING T$;R,O$,H$:LPRINT" ":LPRINT" "
3270 GOSUB 1130 :GOSUB 1160 :LPRINT" # Player"; TAB(28) RIGHT$(T$,82)
3280 GOSUB 1180 :T$ = "__" + STRING$(24,"_") + RIGHT$(T$,84):LPRINT T$
3290 FOR X = 1 TO P:LPRINT USING"\\"; N$(X);                    \"; P$(X):GOSUB 758 :GOSUB 770
3310 GOSUB 1620 :CLOSE:GOSUB 1220 :LPRINT USING T$;A(2)::GOSUB 1600
3320 IF X/3 = INT(X/3) AND X < 20 THEN LPRINT"_"
3330 NEXT X:LPRINT" ":LPRINT STRING$(112,"_")
3340 FOR X=21 TO 22:LPRINT USING "\\"; N$(X);                   \"; P$(X):GOSUB 758 :GOSUB 770
3350 GOSUB 1620 :CLOSE:GOSUB 1220 :LPRINT USING T$;A(2)::GOSUB 1600
3370 LPRINT STRING$(112,"-"):NEXT X:RETURN
3380 REM *** Making corrections ***
3390 CLS:PRINT TAB(22)"Making Corrections":PRINT TAB(21) STRING$(21,131)
3400 PRINT:PRINT" Which of the following do you need to change?"
3410 PRINT STRING$(58,144)
3420 PRINT" 1) Name of school, name of coach, or year (19xx - xx)"
3430 PRINT" 2) Number of wins or losses"
3440 PRINT" 3) Name of player or his jersey number."
3450 PRINT" 4) Name of an opponent (the school name)"
3460 PRINT" 5) Date of a game, location of a game (H/A), or score of game"
3470 PRINT" 6) Statistics for a player"
3480 PRINT" 7) Statistics for an opponent"
3490 PRINT" 8) No changes"
3500 PRINT:PRINT"Please enter your choice (1 to 8):";
3510 T$=INKEY$:T=VAL(T$):IF T<1 OR T> 8 THEN 3510
3520 ON T GOSUB 3540 , 3620 , 3680 , 3850 , 4090 , 4370 :RETURN
3530 REM ***** Change school, coach, or year *****
3540 CLS:GOSUB 270  :PRINT:PRINT"School Name is " S$:GOSUB 1400 :PRINT
3550 IF T$="Y" THEN LINE INPUT"Type in the correct name: ";S$
3560 PRINT:PRINT"Coach's name is "C$:GOSUB 1400 :PRINT
3570 IF T$="Y" THEN LINE INPUT"Type in the correct name: ";C$
3580 PRINT:PRINT:PRINT"The year is "Y$:GOSUB 1400 :PRINT
3590 IF T$="Y" THEN LINE INPUT"Type in the correct year (19xx - xx): ";Y$
3600 GOSUB 250  :RETURN
3610 REM ***** Change number of wins or losses *****
3620 CLS:GOSUB 270  :PRINT:PRINT"Number of wins is ";W:GOSUB 1400 :PRINT
3630 IF T$="Y" THEN LINE INPUT"Enter correct number of wins: ";T$:W=VAL(T$)
3640 PRINT : PRINT"Number of losses is "L:GOSUB 1400 :PRINT
3650 IF T$="Y" THEN LINE INPUT"Enter correct number of losses: ";T$:L=VAL(T$)
3660 GOSUB 250  :RETURN
3670 REM ***** Change player or jersey number  *****
3680 GOSUB 3690 :GOTO 3780
3690 GOSUB 270  :GOSUB 440  :CLS
3700 PRINT"Here is a list of players (with jersey numbers):":PRINT
3710 FOR X=1 TO P STEP 2:PRINT X;P$(X);" (#";N$(X);")",;(P>X)*(" (#";N$(X+1);")")"
3720 IF X+P THEN PRINT TAB(32) X+1; P$(X+1);" (#";N$(X+1);")"
3730 NEXT X:PRINT:PRINT"Which of the above needs to be changed?"
3740 PRINT"Type a number from 0 to"P"(0 means no change)";
3750 T=-1:INPUT T
3760 IF T<0 OR T>P THEN 3750
3770 RETURN
3780 IF T=0 THEN RETURN
3790 CLS:PRINT"Do you wish to change "P$(T)"'s name (Y/N)?":GOSUB 210
3800 IP T$="Y" THEN LINE INPUT"Type in the correct name: ";P$(T)
3810 PRINT:PRINT"Do you wish to change "P$(T)"'s jersey (#"N$(T)") (Y/N)?"
3820 GOSUB 210  :IP T$="Y" THEN LINE INPUT"Type in the correct number: ";N$(T)
3830 GOSUB 410  :RETURN
3840 REM *** Change name of opponent, date, or location ***
3850 GOSUB 270  :OPEN"I",3,"GAMES/TXT":OPEN"O",2,"TEMPFILE/TXT"
3860 FOR X=1 TO G:INPUT#3,O$,D$,H$,SU,ST:CLS:PRINT#90,"Game #";X
3870 PRINT#2288,"Opponent's Name: ":PRINT#2226,O$;:PRINT#4272,"Date (mm/dd/yy):";
3880 PRINT#2298,D$;:PRINT#2336,"Location (H/A):";:PRINT#2354,H$;
3890 PRINT#4440,"Our Score:";:PRINT#2417,SU;:PRINT#4464,"Opponent's Score:";
3900 PRINT#441, ST;:PRINT#579, STRING$(58,140);
```

Listing continued

Listing continued

```
3910 PRINT@718,CHR$(31);"Do you wish to change the opponent's name?"
3920 GOSUB 210
3930 IF T$="Y" THEN LINE INPUT"Type in correct name: ";O$
3940 IF LEN(O$)>14 THEN PRINT@768,CHR$(31)"Name must be less than 15 characters."
     : GOTO 3930 ELSE PRINT@226,O$;"                "
3950 PRINT@718,CHR$(31);"Do you wish to change the date (Y/N)?":GOSUB 210
3960 IF T$="Y" THEN LINE INPUT"Type in correct date: ";D$:PRINT@238,D$;
3970 PRINT@718,CHR$(31)"Do you wish to change the location (Y/N)?":GOSUB 210
3980 IF T$="Y" THEN LINE INPUT"Type in Home or Away: ";H$:PRINT@354,H$;
3990 PRINT@718,CHR$(31)"Do you wish to change our score (Y/N)? ":GOSUB 210
4000 IF T$="Y" THEN INPUT"Our correct score";SU : PRINT@417,SU;;
4010 PRINT@718,CHR$(31)"Do you wish to change the opponent's score (Y/N)?"
4020 GOSUB 210 :IF T$="Y" THEN INPUT"Opponent's new score";ST:PRINT@481,ST;
4030 PRINT@718,CHR$(31);GOSUB 200 :IF T$="N" THEN 3910
4040 PRINT@2,O$;","D$;","N$;","SU;ST:NEXT X:CLOSE
4050 OPEN"I",2,"campfile/txt":OPEN"O",3,"GAMES/TXT"
4060 FOR X=1 TO G:INPUT#2,O$,D$,H$,SU,ST:PRINT#3,O$;",";D$;",";H$;",";SU;ST
4070 NEXT X:CLOSE:RETURN
4080 REM ***** Change statistics for an Individual *****
4090 GOSUB 3650 :PN=T:IF PN=0 THEN RETURN
4100 GOSUB 1560
4110 PRINT:PRINT:PRINT"Choose the number of the game in which "P$(PN)"'s"
4120 PRINT"statistics need to be changed."
4130 INPUT GN:IF GN<1 OR GN>G THEN 4130
4140 CLS:X=PN:R=GN:GOSUB 4150 :GOTO 4290
4150 GOSUB 770 :GOSUB 1620 :CLOSE:GOSUB 1420 :GOSUB 360
4160 GOSUB 490
4170 PRINT@704,CHR$(31);"Now type in the correct amount for each category"
4180 PRINT"or just press <ENTER> to keep the same amount."
4190 PRINT@968, "Press <ENTER> to continue";
4200 IF INKEY$<> CHR$(13) THEN 4200
4210 FOR X1=2 TO 20:N$(X1)="":NEXT X1
4220 N$(2)="Quarters" : N$(3)="Field Goals Made"
4230 N$(4)="Field Goals Attempted" : N$(6)="Free Throws Made"
4240 N$(7)="Free throws Attempted" : N$(11)="Offensive Rebounds"
4250 N$(12)="Defensive Rebounds" : N$(15)="Turnovers"
4260 N$(16)="Assists" : N$(17)="Draw Offensive Fouls"
4270 N$(18)="Steals" : N$(19)="Blocked Shots"
4280 N$(20)="Fouls":RETURN
4290 FOR Y1=2 TO 20:T=A(Y1):IF N$(Y1)="" THEN 4320
4300 PRINT@896,CHR$(31)N$(Y1);:INPUT T
4310 IF T<>A(Y1) THEN S(Y1)=T-A(Y1):A(Y1)=T:T(PN,Y1)=T(PN,Y1)+S(Y1):
     T(21,Y1)=T(21,Y1)+S(Y1)
4320 NEXT Y1:X=PN:GOSUB 670 :GOSUB 770 :X=21:G=GN:GOSUB 880 :X=21
4330 GOSUB 670 :GOSUB 310 :X=21:G=GN:GOSUB 750 :GOSUB 770 :GOSUB 1620
4340 FOR Y1=2 TO 20:A(Y1)=A(Y1)+S(Y1):NEXT Y1:GOSUB 710 :GOSUB 880
4350 GOSUB 880 :RETURN
4360 REM ***** Change Opponent's Statistics *****
4370 GOSUB 270 :GOSUB 1560
4380 PRINT:PRINT:PRINT"Choose the number of the game in which the opponent's"
4390 PRINT"statistics need to be changed."
4400 INPUT GN:IF GN<1 OR GN>G THEN 4400
4410 CLS:X=22:R=GN:PN=22:P$(22)="Opponent":GOSUB 4150
4420 FOR Y1=2 TO 20:T=A(Y1):IF N$(Y1)="" THEN 4450
4430 PRINT@896,CHR$(31) N$(Y1);:INPUT T
4440 IF T<> A(Y1) THEN S(Y1)=T-A(Y1):A(Y1)=T:T(22,Y1)=T(22,Y1)+S(Y1)
4450 NEXT Y1:X=22:GOSUB 670 :GOSUB 770 :GOSUB 880 :G=GN:GOSUB 880
4460 GOSUB 310 :RETURN
4470 REM ***** Add a player to the roster *****
4480 CLS:PRINT TAB(19) "Add a player to the roster" : PRINT : PRINT
4490 PRINT"Do you really wish to add a player (Y/N)?":GOSUB 210
4500 IF T$="N" THEN RETURN
4510 GOSUB 270 :IF P=20 THEN PRINT"No more players may be entered.":RETURN
4520 GOSUB 290 :GOSUB 360 :P=P+1:GOSUB 250
4530 PRINT"Enter name of player #"P"(up to 20 letters)" ; INPUT P$(P)
4540 PRINT:PRINT"Enter jersey number for "P$(P)":: INPUT X$(P):GOSUB 280
4550 IF T$="N" THEN CLS:GOTO 90
4560 GOSUB 310 :GOSUB 410 :GOSUB 510 :X=P:GOSUB 750 :GOSUB 770
4570 FOR X1 = 1 TO G:GOSUB 880 :PUT 1,X1:NEXT X1:CLOSE:RETURN
4580 REM ** Initialize files **
4590 CLS:PRINT TAB(18)"Initialize Team Information": PRINT: POKE 16916,2
4600 PRINT"   Information may be kept for one team only. If you have"
4610 PRINT"information about a team which you wish preserved, press"
4620 PRINT"<BREAK> immediately. Then put this program on another disk,"
4630 PRINT"and begin again.":GOSUB 160
4640 PRINT"   Please enter the following information:"
4650 PRINT@261,"Name of School";:INPUT S$:PRINT@389,"Name of Coach";:INPUT CS
4660 PRINT@ 517,"Year (19xx - xx)";:INPUT Y$:GOSUB 280
```

Listing continued

See our 4★ Review on page 119.

Listing continued

```
4670 IF T$="N" THEN CLS: GOTO 4640
4680 CLS:PRINT@133,"Now you will enter each player's name (up to"
4690 PRINT"20 letters) and each player's jersey number."
4700 P=P+1
4710 PRINT@323, CHR$(31);:PRINT"Name of player #"P;:INPUT P$(P)
4720 PRINT@ 451, "Name will be printed as ";LEFT$(P$(P),20)
4730 PRINT@579, "Jersey Number for ";LEFT$(P$(P),20);: INPUT N%(P):GOSUB 280
4740 IF T$="N" THEN 4710
4750 IF P=20 THEN PRINT@968,"No more players may be entered.";:GOTO 4790
4760 IF P<5 THEN 4700
4770 PRINT@ 960, CHR$(31);"Are there any more players?";:GOSUB 210
4780 IF T$="Y" THEN 4700
4790 POKE 16916,0:G=0:W=0:L=0:GOSUB 250 :GOSUB 290 :GOSUB 310 :GOSUB 410
4800 RETURN
4810 REM ** Instructions **
4820 CLS:PRINT" This program keeps a record of various basketball"
4830 PRINT"statistics. It stores each individual player's statistics"
4840 PRINT"as well as team statistics in 20 categories." : PRINT
4850 PRINT" The first step in using the program is to enter information"
4860 PRINT"about your team - name of school, name of coach, names of"
4870 PRINT"players, etc. The program accommodates up to 20 players." : PRINT
4880 PRINT" Once this has been done, you may enter statistics for a"
4890 PRINT"particular game. You may just press the <ENTER> key to type in"
4900 PRINT"a zero for any category. After entering one player's"
4910 PRINT"statistics, you will be asked if the numbers just entered are"
4920 PRINT"correct. If any are incorrect, you will be asked to enter the"
4930 PRINT"information again.":GOSUB 160
4940 PRINT" The most important part of the program for the coach (and"
4950 PRINT"for the player, too) is the printout of the statistics. This"
4960 PRINT"program prints the statistics in 110 columns. You will need"
4970 PRINT"a printer with this capability. You must either use 11 x 14 in."
4980 PRINT"paper (if your printer prints 132 columns), or you must use"
4990 PRINT"condensed printing (for dot-matrix printers), or you must use"
5000 PRINT"elite printing (for daisy wheel printers). Feeding single"
5010 PRINT"sheets of 8 1/2 x 11-inch paper sideways makes excellent"
5020 PRINT"printouts on a daisy wheel." : PRINT:GOSUB 160
5030 PRINT" You may make the following printouts:"
5040 PRINT"_____"
5050 PRINT: PRINT TAB(5)"1) your team record" : PRINT
5060 PRINT TAB(5)"2) your team totals in which each opponent is listed"
5070 PRINT TAB(10) "along with your team statistics for that game"
5080 PRINT : PRINT TAB(5) "3) your team totals in which each player is listed"
5090 PRINT TAB(10)"along with his totals" : PRINT
5100 PRINT TAB(5) "4) statistics for an individual player" : PRINT
5110 PRINT TAB(5) "5) statistics for a particular game":GOSUB 160 :RETURN
5120 REM ** Housekeeping **
5130 DIM P$(22),S(20),N$(22),T(22,20):DEFINT G,W,L,R,X,Y
5140 P=0:RETURN
5150 REM ** Opening Display **
5160 CLS:FOR Y= 130 TO 898 STEP 64:PRINT@ Y, CHR$(191);:NEXT Y
5170 PRINT@ 899, STRING$(56,176);:FOR X=6 TO 14:SET(X,18):NEXT X
5180 FOR Y=7 TO 12:SET(9,Y):NEXT Y
5190 SET(6,9):SET(7,9):SET(8,9):SET(11,11):SET(13,11):SET(12,12)
5200 FOR Y=28 TO 35:SET(105,Y):SET(106,Y):NEXT Y
5210 FOR Y=36 TO 39:SET(104,Y):SET(107,Y):NEXT Y
5220 FOR Y=39 TO 43:SET(103,Y):NEXT Y
5230 SET(100,39):SET(100,40):SET(109,40):SET(110,40):SET(111,40)
5240 FOR X=100 TO 103:SET(X,31):NEXT X
5250 SET(104,30):SET(104,29):SET(103,29):SET(102,29):SET(101,28)
5260 SET(100,28):SET(104,27):SET(105,26):SET(106,26):SET(107,27)
5270 T$="BASKETBALL STATISTICS"
5280 FOR X=1 TO 22:GOSUB 5330 :PRINT@ 202, LEFT$(T$,X);:NEXT X:T$ ="by"
5290 FOR X=1 TO 2:GOSUB 5330 :PRINT@ 340, LEFT$(T$,X);:NEXT X
5300 T$="David Pleacher"
5310 FOR X=1 TO 14:GOSUB 5330 :PRINT@462, LEFT$(T$,X);:NEXT X
5320 FOR X=1 TO 800:NEXT X:RETURN
5330 T=0:FOR Y=1 TO 18:PRINT@ T," ";:READ T:PRINT@ T, MID$(T$,X,1);
5340 FOR Y1 =1 TO 2:NEXT Y1:RESTORE:PRINT@ 133," ";:RETURN
5350 DATA 689,753,817,881,753,689,624,559,493,427,295,228,160,92,21,16,74,133
5360 REM ** Error Handling Routine **
5370 IF ERL=360  OR ERL=270  OR ERL=440  OR ERL=2650  OR ERL=3040  THEN PRINT
     "file has not been initialized.";PRINT"Error in line"; ERL:GOSUB 160:
     RESUME 70
5380 IF ERL=1020 THEN OPEN"O",3,"GAMES/TXT":RESUME 1030
5390 IF ERL=3160 THEN PRINT"No games have been played.";:GOSUB 160  :RESUME 70
5400 PRINT "Error in line" ERL:GOTO 120
```

End

Continued from p. 63

Program Listing 2. Prism Ring.

```
10 'PRISMATIC RING
15 '12 seconds to execute
20 VIEW(0,0)-(639,239) 'reset entire to viewport
30 CLR:SCREEN 0  ' clear screen and go to graphics screen
40 X=30:Y=30:A=59.70:B=53.50:PI=3.14159
50 AN=19:SZ=24
70 FOR Z=A TO B STEP  -PI/(X+.1)
80 I=330+Y*(5.6667*SIN(Z)):N=130+Y*COS(Z-AN) 'put viewports in
   circle
90 IF Y>83 THEN 120  ' if ring completed go to holding loop
100 VIEW(I,N)-(I+SZ,N+SZ),,1:CLR 'set viewport for size and place-
    ment
110 NEXT Z
120 IF INKEY$="" THEN 120   'hold graphics screen (SCREEN 0)
130 '****  Variables  ****
140 'AN is angle of ring --causes different deaigns by choosing
    numbera between 10 and 360
150 'X is size of space between boxes (viewports)-smaller number
    makes larger spaces
160 'Y is size of ring-smaller number makes smaller ring
170 'A is beginning of loop (to make one complete ring)
180 'B is end of loop
190 'SZ is size of box (viewport)-larger number makes larger boxes
```

End

Program Listing 3. Viewport.

```
10 'EXAMPLE VIEWPORTS WITH TEXT - VIEWPORT/BAS
20 A$=CHR$(&HA0)+CHR$(&H0A)
30 SCREEN 0:CLR 'go to graphics screen and clear it
40 LINE(0,0)-(639,239),,B 'place outlined box around perimeter of
   entire graphics screen
50 PAINT(320,120),A$,1 'paint background on screen
60 VIEW(100,30)-(470,160),0,1 'define first viewport
70 GOSUB 200 'access screen writing subroutine to place data on
   screen in viewport
80 VIEW(200,80)-(550,185),0,1 'define second viewport
90 GOSUB 200 'same subroutine to put data in viewport
100 VIEW(0,0)-(639,239) 'define entire screen to viewport
105 IF INKEY$="" THEN 105
110 END
200 'Subroutine to write data to viewports
210 GLOCATE(1,1),0 'locate coordinates to place data
220 PRINT#-3, "THIS IS THE CURRENT VIEWPORT THAT HAS BEEN DEFINED
BY THIS PROGRAM"
230 RETURN
```

End

Program Listing 4. Circle.

```
10 DIM V%(530) 'use integer to save memory
20 SCREEN0:CLR  'go to graphics screen and clear it
30 CIRCLE(50,50),20 'draw circle on screen
40 LINE(2,2)-(99,79),,B 'draw outline box just inside
parameters of area captured by GET array in line 60
50 PAINT(5,5),1,1 'color inside of box
60 GET(1,1)-(100,80),V% 'store section of screen containing
circle
70 CLR  'clear graphics screen
80 PUT(1,1),v%,PSET 'place circle and box back on screen
90 FOR K=1 TO 2000:NEXT K 'hold picture
```

End

Program Listing 5. Windows.

```
100 'GLEN E. SPARKS
110 '6106 CUSTER
120 'SOUTH ROCKWOOD, MI 48179
130 '
140 ' WINDOWS/BAS
150 '***  Define, initialize and dimension variables ***
160 CLEAR 1000
170 DIM V%(2100):DIM V1%(2100)
180 DIM G1(15),AC$(15),PR(16),AM(15)
190 DIM PT$(15) 'PainT strings
200 PT$(0)=CHR$(&H0A)+CHR$(&HA0)
210 PT$(1)=CHR$(&HAA):PT$(2)=CHR$(&HF0)+CHR$(H0F)
220 PT$(3)=CHR$(&H0A)+CHR$(H04)
230 PT$(4)=CHR$(&B4D):PT$(5)=CHR$(&HF0)+CHR$(&H0F)
240 PT$(6)=CHR$(&HAA)+CHR$(&H3C)
250 PT$(7)=CHR$(&H03)+CHR$(&H0C)+CHR$(&H30)+CHR$(&HC0)
260 PT$(8)=CHR$(&H04):PT$(9)=CHR$(&H3C)
270 PT$(10)=CHR$(&H29):PT$(11)=CHR$(&H19)
280 PT$(12)=CHR$(&HAA):PT$(13)=CHR$(&H2D):PT$(14)=CHR$(&H6F)
290 P2=2*3.14159:R=50
300 XC=100:YC=45:Q$="$$$,$$$.$$"
310 CT=0 'CounT the times graph redone
320 '****  Begin program  ****
330 GOSUB 1520   'build and store hidden menu
340 GOSUB 590  ' pie chart
350 '*** Manipulate screen and data through pull down menu ***
360 IF INKEY$="" THEN 360
370 VIEW(0,0)-(639,239):GET(200,60)-(498,160),V1%
    'CAPTURE AREA UNDER WHERE MENU WILL GO
380 PUT(200,60),V%,PRESET  'superimpose menu
390 CW$=INKEY$:IF CW$="" THEN 390 ELSE 400
400 PUT(200,60),V1%,PSET 'replace area covered by menu
410 CW=VAL(CW$):ON CW GOSUB 510,520,530,540,550,590
    'menu choices
420 IF CW=7 THEN 450  'exit menu and hold window screen
430 IF INXEY$="" THEN 430
440 GOTO 370  'go back through menu
450 IF INKEY$="" THEN 450
460 END
470 '
480 '=========SUBROUTINES=========================
490 '
500 '*** Subroutines to define and clear large windows ***
510 VIEW(3,11)-(210,118),0,1:CLR:RETURN 'window 1
520 VIEW(6,136)-(330,225),0,1:CLR:RETURN 'window 2
530 VIEW(220,24)-(635,110),0,1:CLR:RETURN  'window 3
540 VIEW(335,132)-(635,225),0,1:CLR:RETURN 'window 4
550 SCREEN0:CLS:PRINT"PRINT-OUT OF REPORT -MAKE SURE PRINTER IS ON
"
560 PRINT"OO YOU WISH TO PRINT OUT REPORT NOW (Y or N)"
```

Listing 5 continued

Listing 5 continued

```
570 INPUT P$:IF LEFT$(P$,1)="y" THEN 580  ELSE 370
580 LPRINT CHR$(27);CHR$(20):CND"I","GPRT2" 'set printer to
    condensed and print out chart
590 '**** Pie chart and windows screen ****
600 IF CT>0 THEN TA=0 'set total to 0 for next data/chart
610 '--- Input data ---
620 CLS:PRINT"KEEP TITLE OF REPORT TO 25 CHARACTERS"
630 LINE INPUT"TITLE OF CHART (SALES, INVENTORY, etc.) ";T%$
640 IF LEN(T%$)>25 THEN PRINT"TOO LONG A TITLE":GOTO 630
650 PRINT"PERIOD COVERED BY PIE CHART DATA"
660 LINE INPUT MN$:PRINT
670 IF LEN(MN$)>25 THEN PRINT"TOO LONG":GOTO 650
680 INPUT"NUMBER OF ENTRIES (MAX 9) ";N
690 IFN>9 THEN PRINT"ENTRIES MORE THAN 9-ACCOUNT DATA MAY NOT ALL
    FIT IN WINDOW"
700 IF CT>0 THEN PRINT"LAST WINDOWS CNOSEN ";W1" AND ";W2
710 PRINT"CHOOSE WINDOW FOR ACCOUNTS (1-4) "
720 INPUT W1:IF W1>4 OR W1<1 THEN 710
730 INPUT"CHOOSE WINDOW FOR PIE CHART (1-4) ";W2
740 IF W2>4 OR W2<1 THEN 730
750 IF W1=W2 THEN PRINT"YOU HAVE CHOSEN THE SAME WINDOW-DATA WILL
BE OVERWRITTEN BY THE GRAPH":C1=1
760 PRINT:PRINT"LIMIT ACCOUNT LENGTH TO 9 CHARACTERS"
770 PRINT STRING$(50,"-")
780 '--- Input chart data ---
790 FOR I=1 TO N:PRINT"#";I
800   LINE INPUT"ACCOUNT? ";AC$(I)
810 IF LEN(AC$(I))>9 THEN PRINT"TOO LONG":GOTO 800
820   INPUT"AMOUNT ";AM(I):PRINT
830 NEXT I
840 '--- Calc total and slice size ---
850 FOR I=1 TO N
860   TA=TA+AM(I)
870 NEXT
880 FOR I=1 TO N
890   PR(I)=AM(I)/TA*100
900 NEXT I
910 '--- Window choice and set up screen ---
920 IF CT<1 THEN GOSUB 1310
930 CW=W1:GOSUB 1710 'choose window for title
940 GLOCATE(1,5),0:PRINT#-3,T%$ 'print title
950 CW=W1:GOSUB 1270 'set window for accounts data
960 '---Print out accounts ---
970 X1=1:Y1=5:FOR I=1 TO N
980     GLOCATE(X1,Y1),0:PRINT#-3,"#";I;AC$(I)
990     GLOCATE(X1+120,Y1),0:PRINT#-3, USING Q$;AM(I)
1000    Y1=Y1+9
1010 NEXT
1020 CW=W2:GOSUB 1710 'set window for period covered
1030 GLOCATE(1,5),0:PRINT#-3,MN$ 'print month
1040 CW=W2:GOSUB 1270 'set window for chart
1050 '--- Draw chart on screen ---
1060 CIRCLE(XC,YC),R
1070 GLOCATE(1,5),0:PRINT#-3, USING Q$;TA 'TOTALS OF AMOUNTS
1080 FOR I=0 TO N
1090   AA=P2/100*PR(I)+AA
1100   X0=XC+R*COS(AA)
1110   Y0=YC-R*SIN(AA)*0.5
1120   AB=P2/100*PR(I+1)+AA
1130   G1(I)=(AA+AB)/2
1140   G2=((XC-10)+(R+9)*1.15*COS(G1(I)))
1150   G3=((YC-2)-(R+9)*1.15*SIN(G1(I))*0.5)
```

Listing 5 continued

Listing 5 continued

```
    GLOCATE(G2,G3),0:IP PR(I+1)>1 AND I<N THEN PRINT#-3,I+1
    IP PR(I)>1 THEN LINE (XC,YC)-(X0,Y0)
NEXT I
'--- Paint slices of pie ---
FOR I=0 TO N-1
  XL=XC+R*0.5*COS(G1(I))
  YL=YC-R*0.5*SIN(G1(I))*0.5
  IF PR(I+1)<=1 THEN 1250
  PAINT (XL,YL),PT$(I),1
NEXT I
RETURN
'***Subroutine to choose and clear window for output***
IFCW>4THENCW=1
ON CW GOSUB 510,520,530,540
RETURN
'*** set up initial screen ***
VIEW(0,0)-(639,239):CLS:SCREEN 0
GOTO1350
PAINT(320,120),CHR$(&H0A)+CHR$(&HA0),1
VIEW(3,1)-(114,10),0,1:CLS:GLOCATE(1,5),0:PRINT#-3,"#1"
VIEW(15,1)-(210,10),0,1:CLS 'message window 1
GOSUB510    'clear area of window 1
VIEW(6,124)-(16,134),0,1:CLS
GLOCATE(1,5),0:PRINT#-3,"#2"
VIEW(17,124)-(330,134),0,1:CLS 'message window 2
GOSUB520    'clear window 2
VIEW(2,227)-(637,236),0,1:CLS 'area for spacebar menu
GLOCATE(19,5),0:PRINT#-3,"   ********PRESS SPACEBAR FOR PULL
 DOWN MENU********"
VIEW(220,10)-(234,20),0,1:GLOCATE(10,5),0:PRINT#-3,"#3"
VIEW(235,10)-(635,20),0,1:CLS 'message window 3
GOSUB530    'clear area window 3
VIEW(335,120)-(349,130),0,1:CLS:GLOCATE(1,5),0:PRINT#-3,"#4"
VIEW(350,120)-(635,130),0,1:CLS ' message window 4
GOSUB540    'clear area of window 4
CT=1 'Count set at 1
RETURN
'****Set up hidden pull down menu and store in array****
VIEW(0,0)-(639,239):CLS:SCREEN 0:LINE(2,2)-(300,110),,B
GLOCATE(5,7),0:PRINT#-3,"      _____OPTIONS_____"
LINE(2,15)-(300,15)
GLOCATE(5,10),0:PRINT#-3,"ERASE CONTENTS OF WINDOW #1"
LINE(2,27)-(300,27)
GLOCATE(5,30),0:PRINT#-3,"ERASE CONTENTS OF WINDOW #2"
LINE(2,39)-(300,39)
GLOCATE(5,42),0:PRINT#-3,"ERASE CONTENTS OF WINDOW #3"
LINE(2,51)-(300,51)
GLOCATE(5,54),0:PRINT#-3,"ERASE CONTENTS OF WINDOW #4"
LINE(2,63)-(300,63)
GLOCATE(5,66),0:PRINT#-3,"PRINT SCREEN              #5"
LINE(2,75)-(300,75)
GLOCATE(5,70),0:PRINT#-3,"DATA AND GRAPH            #6
LINE(2,87)-(300,87)
GLOCATE(5,90),0:PRINT#-3,"EXIT PROGRAM              #7"
GET(2,2)-(300,110),V%
RETURN
'*** message windows ***
ON CH GOSUB 1730,1740,1750,1760:RETURN
VIEW(15,1)-(210,10),0,1:CLS:RETURN  'message 1
VIEW(17,124)-(330,134),0,1:CLS:RETURN 'message 2
VIEW(235,10)-(635,20),0,1:CLS:RETURN 'message 3
VIEW(350,120)-(635,130),0,1:CLS:RETURN 'message 4
```

End

1985 ARTICLE INDEX

Articles are alphabetical by author within each category. Listings are in the form: author's last name, article title, issue:page, (debug issue:page), (Computer model numbers) Description.

BUSINESS

Garms, "NovaCalc," 1:82 (7:25), (I, III) Full-featured Basic spreadsheet.
Lake, Golden, and Lett, "Reasonable Deductions," 3:74. Home computer tax deductions explained.
McAnaney, "Savings and Loan," 11:83. (I, III, 4, 1000, 1200) Calculate loan balances by Rule of 78s.
Terry, "Password Bypass," 11:72. (I, III, 4, 1000, 1200) Crack protected files.
Wallace, "Don't Be Late," 6:74 (10:25). (I, III, 4, 1000, 1200) Set up a critical path schedule.

COMMUNICATION

Dixon and Maloney, "On the Boards," 7:36. 1,850 BBS numbers.
Shulman, "The Perfect Host," 9:41. (4). Use Memdisk to transfer data at high baud rates.

GAMES

Lewicki and Karls, "Monster Mashing," 2:50. (III) A Dungeons and Dragons type fantasy.
Zare, "The Pecking Order," 2:59. (I, III) Number-crunching in a game format.

GENERAL

Harmon, "Mail Order Maxims," 10:8. Tips on making mail-order purchases.
Pleacher, "Net Results," 12:52. (I, III, 4) Basketball statistics program.

GRAPHICS

Abbassian and Sparks, "Drawing in Detail," 9:56. (III, 4) MacPaint-style high-resolution graphics.
Graebner and Graebner, "Clear-Cut Trends," 4:40. (III, 4, 1000, 1200) Create detailed high-resolution line and bar graphs.
Justice, "Fractals in Focus," 5:58. (III, 4, 1000, 1200) Produces fractals in high-resolution detail.
Leibow, "Grade-A Graphics," 3:44. (III) A graphics editor that lets you create a screen, combine backgrounds, produce mirror images, plus more.
Reifman, "Picture This," 2:56. (III) A graphics program that lets you create designs with graphics blocks, lines, letters, fill-ins, and more.
Rogerson, "Rembrandt Redux," 12:78. (III) Access screen-save commands from Sparks' Rembrandt graphics editor (9:52) and dump screens to Epson printers.
Smith, "Upgraded Graphics," 8:76. (4) Add the Point, Set, and Reset commands to Basic programs.
Sparks, "Window Screens," 12:58. (III, 4) Create high-resolution windows with BasicG.
Van Beverhoudt Jr., "By the Numbers," 2:68 (7:25). (I, III) Plot numerical data on either line or bar charts; then display the graphs sequentially with slide show option.

SCIENCE

O'Neal, "Formula Solutions," 1:116. (I, III, 4, 1000, 1200) Solving real or complex roots of equations.
Wood, "Next-Generation Software," 10:70. (I, III, 4, 1000, 1200) Re-create Gregor Mendel's genetics experiments.

TECHNIQUE

Goodale, "On the Record," 1:106. (I, III) Combine the best features of sequential and random-access files.

TUTORIAL

Brothers, "Labor Saver," 2:62. A guide to tools for the Assembly-language programmer.
Brothers, "Mutual Understanding," 6:82. (I, III, 4) Describes file formats for different assemblers and tells how to make them compatible.
Brothers, "Language Arts," 7:48. (I, III, 4) A comparison of three commercial Basic authoring systems.
Capps, "Fight Simulator," 5:40. (I, III, 4, 1000) How to design a combat simulator.
Dalton, "Prose and Cons," 11:36. (4) A comparison of eight word processors.
Dixon and Rowell, "Dueling Software," 8:56. (4, 1000) Two *80 Micro* editors pick their ideal software libraries.
Dixon, "You Can Get There from Here," 9:36. How to move files from one computer to another.
Harrell, "Born to Run," 12:38. Tutorial on the C language.
Heid, "The GW Difference," 2:42. (I, III, 4) The differences between GW- and TRS-80 Basic.
Knight, "Summer Romance," 8:38. (4, 4P) Model 4 Basic's enhanced commands and functions.
McGarvey, "Running Hard," 10:44. DOS and hard disk system.
Meyer, "Hard Decisions," 10:40. Things to know before you buy a hard disk.
Meyer, "Printer Preservation," 11:44. How to maintain and repair your printer.
Payne, "That Sinking Feeling," 3:38. A tutorial on disk errors.
Payne, Sr., "The Quiet Command," 11:68. (I, III, 4, 1000, 1200) A little-known Basic command.
Rowell, "The Tandy 1000 Tip Sheet," 6:38. (1000) Advice, tricks, and patches.
Rowell, "Sifting Through GW-Basic," 8:46. (1000) A guide to Model 1000 Basic.
Rowell, "Have Data, Will Travel," 10:52. Running software written for one computer on another.
Welch, "Date Lines," 4:72. (III) Sorting dates with Profile III Plus.
West, "Current Events," 5:62. A consumer's guide to surge protectors.

UTILITY

Anaya, "Restricted Entry," 5:70. (I, III, 4, 1000, 1200, 2000) Restrict program input.
Boggs, "Stationary Department," 5:74. (4) A scroll-protection utility.
Boggs, "Files From the Crypt," 10:58. (4) Recover killed files.
Brothera, "My 10 Favorite Assembly-Language Subroutines," 9:48. (III, 4) Input and screen-handling routines to merge with your Basic programs.
Dyke, "The Right Address," 12:74. (4) A program that locates memory addresses under TRSDOS 6.X.
Goodwin, "Getting Ahead," 7:65. (III) Type-ahead feature for TRSDOS 1.3 and Cassette Basic.
Hunter, "Room Available," 6:60. (I, III) Increase disk storage capacity for text files with Huffman code.
Knight, "Model 4 Scripsit the Write Way," 1:60. (4) A package of Scripsit enhancements.
Knight, "Key Notes," 10:62. (4, 4P) Program the 4's function keys.
Levinson, "Patch Work," 1:112. (3:27, 8:25). (III) Thirteen patches to TRSDOS 1.3.
Levinson, "Patch Work II: The Sequel," 8:72. (III) More patches to TRSDOS 1.3.
Levinson, "Merge Right," 11:54. (I, III) Merge non-ASCII files with Basic programs.
Oler, "Interrupt Anytime," 12:66. (III) Interrupts for TRSDOS 1.3.
Pavlicek, "Full Recovery," 6:57. (4) Recover lost SuperScripsit files.
Pelzer, "A Basic Programmer's Best Friend," 9:66. (4, 4P) Cross-reference Basic key words; variables; and GOSUB, GOTO, and Restore statements by line number.
Quindry, "Running Like the Wind," 1:42 (5:29). (I, III) A Basic compiler.
Risler, "Extra-Strength DOS," 8:48. (III, 4) Soup up LDOS 5.X/TRSDOS 6.X.
Robinson, "Patch Maker," 10:76. (III) Install patches with a Build file.
Schweim, "A La CRT," 3:58 (7:26). (I, III, 4) Customized main menus.
Smith, "A Sort Story," 3:70. (4) Fast string sort for Model 4 Basic.
Smith, "Run-O-Matic," 7:78. (4) Disk menus that let you load programs with the touch of a key.
Snyder, "The Great Divide," 8:62. (4, 4P) Partition your 128K system to switch between two programs in Model III mode.
Sparks, "Picture Perfect," 1:98. (I, III, 4) Complete graphics production package.
Wagner, "Fast Figures," 11:60. (I, III) Binary I/O zips up disk accesses and saves disk space.
Williams, "Zap Master," 4:62. (8:25). (4) Refined Debug utility.
Wilson, "Made-to-Order Sorts," 7:70. (I, III, 4, 1000, 1200) Customized Basic sort generator for sequential files.
Wolcik, "Storage to Spare," 6:54. (4) Increase SuperScripsit's storage space.
Wolfskill, "Total Recall," 7:54. (10:25) (4) Create and display help screens.
Zenzel Jr., "Write Away," 12:41. (4, 1000) Basic C interpreter.

1985 REVIEW INDEX

Reviews are listed alphabetically by product within each category, followed by the manufacturer's name and issue and page number.

BOOKS

How to Get the Most Out of CompuServe, Bantam Books, 1:152.
Mod-4 by Jack, Crest Software, 8:115.
The Source: TRSDOS/LS-DOS 6.2, Logical Systems Inc., 1:158.
Teach Your TRS-80 to Program Itself, Tab Books Inc., 9:100.
Inside CP/M Plus: A Guide for Users, Holt, Rinehart, and Winston, 9:103.
Learning Basic for the Tandy 1000/2000, Compusoft Publishing, 10:111.
Programmer's Guide to LDOS/TRSDOS Version 6.X, Misosys, 11:119.
How to Use Your Radio Shack Printer, Tandy/Radio Shack, 12:126.

GAMES

Chess Classics, Noteworthy Software, 10:36.
Crossword Master, Just Software, 4:119.
Flight Simulator, Microsoft Corporation, 7:114.
Tournament Chess, Rapidynamic Software Inc., 10:36.

HARDWARE

Grafyx Solution, Micro-Labs Inc., 11:119.
Joy-Mouse Interface, Micro-Labs Inc., 12:29.
Lucid, Portable Computer Support Group, 4:31.
Tandy 1000, Tandy/Radio Shack, 4:50.
Tandy 1200 HD, Tandy/Radio Shack, 3:31.
Tandy 200, Tandy/Radio Shack, 7:29.

LANGUAGES

CP/M 2.2, Montezuma Micro, 3:35.
BetterBasic, Summit Software Technology, 6:33.
GBasic 3.0, Micro-Labs Inc., 12:29.

PRINTERS

CGP-220, Tandy/Radio Shack, 1:36.
Hush 80, Ergo Systems Inc., 6:34.
ThinPrint 80, Axonix Corporation, 8:34.
ThinType, Axonix Corporation, 8:34.

SOFTWARE

Accounts Receivable 1.85, Holman Data Products, 9:31.
Act III, Avanti Associates, 4:36.
Backrest, Powersoft, 10:35.
Brainstormer, Soft Path Systems, 9:100.
Business Graphics Analysis Pak, Tandy/Radio Shack, 10:31.
Checkbook Plus, H&E Computronics, 2:38.
DeskMate, Tandy/Radio Shack, 5:31.
Disk Term, Indiana Software Co., 4:29.
DotWriter 4.0, Prosoft, 6:29.
Draw, Micro-Labs Inc., 12:29.
EasyWriter 1, Information Unlimited Software, 5:116.
Electric Desk, Alpha Software Corporation, 6:114.
Electric Webster, Cornucopia Software Inc., 3:116.
Etch-A-Mouse, Soft Horizons, 6:114.
FasType, Press A Software, 11:119.
Fullview, Mitek Systems, 2:132.

HomeworD, Tandy/Radio Shack, 9:102.
Home Health Guide for Children, Clinical Reference Systems Inc., 10:110.
IDEA!, Traveling Software, 9:102.
Intercalc, Comtrol Data Products, 10:34.
Interprinter, Comtrol Data Products, 10:34.
Let's Talk, Good Software Co., 2:35.
List/Key, The Soft Place, 2:130.
Little Brother, Logical Systems Inc., 9:34.
Math Master Series, PAB Software Inc., 8:113.
Microsoft Word 2.0, Microsoft Corporation, 8:114.
MLink Communications System, Corporate Microsystems Inc., 10:113.
Money Decision Series, Tandy/Radio Shack, 12:33.
Monte's Window, Montezuma Micro, 8:36.
Offix Personal Office System, Emerging Technologies, 9:102.
Omniterm2, Lindbergh Systems Inc., 7:114.
Orchestra-90, Tandy/Radio Shack, 3:114.
PFS:File and Report, Tandy/Radio Shack, 7:33.
PowerScript, Powersoft, 7:114.
Producer, Software of the Future, 1:39.
Profile 4 Plus, Tandy/Radio Shack, 11:114.
PRO-NTO, Misosys Inc., 11:29.
Remote Control, Kensington Microware Ltd., 6:113.
Super DBM, Scientific Analysis, 6:34.
Superlog 4, KSoft Inc., 8:29.
Tax-Prep, EZWare Corporation, 4:111.
Telecommuter, Sigea Systems Inc., 12:126.
TK!Solver, Tandy/Radio Shack, 7:115.
T/Maker (CP/M), T/Maker Company, 1:34.
T/Maker 4.03 (TRSDOS), T/Maker Company, 10:111.
Total Learning System, Total Learning Systems Inc., 10:110.
TRSDOS 6.X Training Course, Tandy/Radio Shack, 6:115.
Typitall, Howe Software, 12:125.
Volkswriter Deluxe 2.0, Lifetree Software Inc., 5:110.
Window-Comm, Software Consultants, 11:31.
Windowpad, En Fleur Corporation, 10:34.
Word MachInc 3.0, Pel-Tek, 7:113.
Word Perfect 4.0, Satellite Software Int., 12:125.
xT.CAD, Microdex Corp., 2:37.

UTILITIES

BEEP, Logical Systems, 7:34.
CopyIIPC, Central Point Software Inc., 11:119.
Convert 3 to 2000, Educational Micro Services, 5:114.
DISnDATa, Pro/Am Software, 2:128.
DoubleDuty, Tandy/Radio Shack, 10:36.
DSM4, Logical Systems Inc., 11:116.
GW-Basic Compiler, Tandy/Radio Shack, 8:30.
Hypercross (Supercross/XT), Hypersoft, 9:29.
Hyperzap, Hypersoft, 12:34.
LS-Utility Disk, Logical Systems, 5:35.
Multi-Basic Compiler, Alcor Systems, 11:33.
MULTIDOS 80/64, Alphabit Communications, 12:124.
Pro-Create, Misosys, 8:114.
PRO-X-FTS, Misosys, 12:128.
Overdrive, Logical Systems, 6:114.
NEWDOS/80 Utilities, Software Success, 6:31.
Norton Utilities 3.0, The Norton Utilities, 8:114.
Rapidos, Rapidynamic Software Inc., 7:113.
Supercross/XT, Powersoft, 9:29.
Supermod4, Intellitech Corporation, 9:33.
Vtvace, Wittsoft Inc., 4:115.
WindowDOS, WutdowDOS Associates, 11:118.
Zeus Editor/Assembler, Cosmopolitan Electronics Co., 1:154.

1985 LOAD 80 INDEX

Each entry lists the article name, page number, and names of major programs.

JANUARY

"Running Like the Wind" (42); FastBas, Pong. An updated Basic compiler.
"Model 4 Scripsit the Write Way" (60); ScripAid. An enhancement package for Model 4 Scripsit.
"NovaCalc" (82); NovaCalc. A Basic spreadsheet program.
"Picture Perfect" (98); CHARGEN. Create your own graphics characters.
"On The Record" (106); RANDISK. Create and read indexed random-access disk files.
"Formula Solutions" (118); Cubic. Solve cubic, quadratic, and linear equations.
Project 80 (120); Interrupt. Use the 8259A interrupt controller with an I/O board.
BBS Express (132); BBS1, BBS2. A routine for locating message numbers.
The Next Step (140); Demo. Demonstration program to create a beep generator.

FEBRUARY

"Picture This" (56); Graph. Create pictures and combine up to nine screens for a composite illustration.
"The Pecking Order" (59); NOEATERS. Gobble up the numbers on the screen to win the game.
"By the Numbers" (68); GraphMaster. Plot or display data in line or bar charts.
The Next Step (112); Demo. Use DOS exits to add commands to Basic.
BBS Express (104); BBS9. Sort-and-search program for your BBS.
Bonus program: Mail List. Model 4 mailing list.

MARCH

"Grade-A Graphics" (44); Graph. Versatile graphics generator.
"A La CRT" (58); Listings 1–12. Basic subroutines that create menus.
"A Sort Story" (70); Sort. A string sort for Model 4 Basic.
BBS Express (88); BBS. This module gets your bulletin board up and running.
The Next Step (100); Helpfile. A help file generator program.
Bonus program: Alien Shuffle. Rearrange aliens into proper groups.

APRIL

"Clear-Cut Trends" (40); Grapher. Draw and print out high-resolution line and bar graphs.
BBS Express (90); BBS. The BBS data base catalog module.
"Zap Master" (62); Disk Zapper. Read and modify Model 4 disk sectors.
The Next Step (98); Filter. A video filter program.
Bonus program: TapeDisk. Tranfers all files from the Load 80 cassette to disk.

MAY

"Fight Simulator" (40); Endgame. Combat simulation.
"Reatricted Entry" (70); Prompter. Predefine acceptable user input.
"Stationary Department" (74); Scroll. Adds scroll protection to Model 4 video display.
"Fractals in Focus" (58); Fractal3, Fractal4. High-resolution fractals.
BBS Express (92); BBS. Put the finishing touches on your bulletin board system.
The Next Step (100); Extend. Demonstration of an extended command interpreter.
Bonus program: Convert. Convert Apparat ur Series 1 source files to ALDS format.

JUNE

"Extra-Strength DOS" (48); Setup. Add features to LDOS 5.X.X or TRSDOS 6.X.X.
"Full Recovery" (57); Repair. A file-recovery program for Model 4 SuperScripsit.
"Room Available" (60); Compress, Decompress. Conserve disk space using Huffman format.
"Don't Be Late" (74); CPS. Develop timelines for projects.
The Next Step (102); Driver, DRIVETST. Add windowing capability to the Model 4
Spreadsheet Beat (110); Documenter. Print SYLK files in an easy-to-read format.
Bonus program: Convert2. Convert source file formats.

JULY

"Total Recall" (54); Helper. Display help screens on the Model 4
"Getting Ahead" (64); Type. A type-ahead utility for TRSDOS 1.3.
"Made-to-Order Sorts" (70); Sorter. A custom sort generator.
"Run-O-Matic" (78); Autorun. Load Model 4 disk files from a menu.
Project 80 (84); Display. Remote display board controller.
The Next Step (100); WS, Test, WD. Part II of the windowing system.

AUGUST

"Summer Romance" (38); Main Menu, Clock. Menu sequencing and clock display routines.
"The Great Divide" (62); Times2. Partition your 128K Model 4 and switch between two programs in Model III mode.
"Patch Work II: The Sequel" (72); Domaker. Install up to 13 TRSDOS 1.3 patches.
"Upgraded Graphics" (76); Pixel. Add the commands Point, Set, and Reset to Model 4 Basic.
The Next Step (98); WU. Window system user interface, Part III.
Spreadsheet Beat (108); BondYTM. Calculate bond yield-to-maturity using VisiCalc.

SEPTEMBER

"My 10 Favorite Assembly-Language Subroutines" (48); List1, List2, List3, List4, List5. Ten input and screen-handling subroutines.
"A Basic Programmer's Best Friend" (66); Crosscheck, Keywords. Model 4 Basic cross-reference program.
"Drawing in Detail" (56); Rembrandt, Screengrid, Hexagon, Computer. High-resolution icon-driven graphics editor.
The Next Step (90); Line. A line-drawing subroutine and sample Basic program.
Spreadsheet Beat (94); Report Card. Calculate students' grades with VisiCalc.

OCTOBER

"Files from the Crypt" (58); Restore. Retrieve killed files on the Model 4
"Key Notes" (62); PF/FLT. Model 4 function key filter.
"Next-Generation Software" (70); Genetics. Recreate genetic experiments.
"Patch Maker" (76); Patcher. A patch management program.
The Next Step (100); File Chop. Condense random-access files.
Spreadsheet Beat (106); TaxEst85. Track expenses and estimate federal income tax with Multiplan.
Bonus program: Diskbug, TRSDOS 1.3 debug utility.

NOVEMBER

"Merge Right" (54); Merge. Merge non-ASCII files on the Models I and III.
"Fast Figures" (60); DISQUICK. Faster reads and writes with binary I/O.
"Savings and Loan" (83); Loan. Calculate interest by the rule of 78s.
"Password Bypass" (72); UNPROT. A Model I/III/4 file protection remover.
The Next Step (102); List1. Strip trailing blanks, encode and decode strings.
Spreadsheet Beat (108); Budgeter. Maintain a monthly personal budget with Multiplan.
A program update: REMBRAN4. Updated version for the Model 4; first appeared on September 1985, Load 80.
Bonus program: SpeedDOS. 4MHz operation in the Model III mode.

DECEMBER

"Write Away" (41); C Trainer. Basic C interpreter.
"Net Results" (52); Hoops. Basketball statistics program.
"Window Screens" (58); Sinewave, PrismRing, Windows. High-resolution graphics and pie chart application.
"Interrupt Anytime" (66); Break In, Scroll. Programmable interrupts for TRSDOS 1.3.
"The Right Address" (75); Locator. Finds TRSDOS 6.X system addresses.
"Rembrandt Redux" (76); List1, List2. Screen dump routines for our Rembrandt graphics editor.
Tidbit #30 (83); Page. File list utility.
Project 80 (85); Convert. Converts object files to hex/ASCII.
The Next Step (108); Squeeze. Condenses Model 4 programs.
Bonus program: Delete. Multiple LDOS 5.X kill command.

Tandy 1000

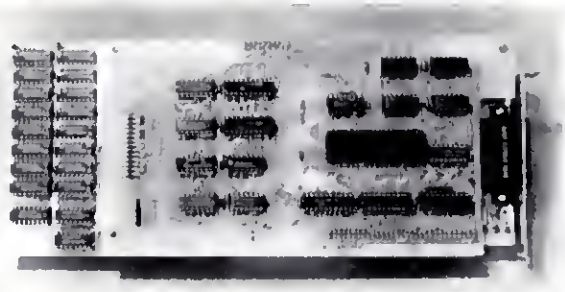
Circle 455 on Reader Service card.

TanPak™

The Ultimate Expansion for the Model 1000!

The TanPak expansion board has been designed to allow expansion beyond the scope of the standard Model 1000. Seven of the most needed functions/features have been combined into one package using only one expansion slot. Your remaining spaces are left free for future expansion needs. And the best part of all, it will save you money over the Tandy Boards.

TanPak™ O.k. $349



Functions Include:
DMA
Serial Port
Memory up to 512K
Clock/Calendar
Printer Spooler
Memory Disk
Expansion Port for Future Options

TanPak features include.
Gold Edge Cards
Supporting Software
Pre-Tested and Burned In
Full Documentation
One Year Warranty
Expandable to 512K

TanPak™ Secondary

Made to work with a Tandy 1000 memory board as the second memory board. Featuring a Serial Port, Clock/Calendar, Memory Expansion to 256K, Printer Spooler, and Memory disk

TanPak™ Secondary ..... **$249.**

Tandy 1000 Hard Drives

HDS Hard Drives for the Tandy 1000 allow booting directly from the Hard Drive using the Tandy DOS. All units are complete with controller, ready to plug in and use. The Internal Hard Drive Units replace the top disk drive inside the 1000, or request an external unit for only $150. more.

10 Meg Internal ..... **$549.**
20 Meg Internal ..... **$749.**
43 Meg Internal ..... **$1399.**

TanTel

Internal 300/1200 Baud Modem **$249**

8087 Board

8087 Math Co-processor board for the Model 1000.
Plugs in internally but does not use one of the 3 expansion slots
..... $249.

(Available Fall 85')

Model 1, 3, 4 Hard Drives

Model 1, 3, 4 Hard Drives

	Primary	Secondary
5 Megabyte	$795.	$595.
10 Megabyte	$1095.	$895.
20 Megabyte	$1295.	$1095.
30 Megabyte	$1895.	$1695.

(Model 1 add $50)

Hard Disk Subsystem Features:

- All sizes rated after formatting.
- Your choice of DOSPLUG, LDOS or TRSDOS 6 x Drivers included.
- NEWDOS 80/Version 2.5 Systems Available.
- Up to two secondary drives may be added.
- Error checking and correcting controller.
- Buffered seek drives for improved access time.
- Built in power up diagnostics.
- Plated media for long disk life.
- Heavy duty power supply
- Gold connectors used throughout
- 1 year warranty

HDS Multiplexer

Allows 4 Computers to use a HDS Hard Drive,
Base unit ..... $1295.

Model 3/4 RS232 Kit ..... **$69.**

Hard Drive Specialist

16208 Hickory Knoll
Houston, Texas 77059
1-713-480-6000
orderline
1-800-231-6671

Ordering Information
Use our WATS line to place your order via Visa, MasterCard, or Wire Transfer. Or, you can mail your payment directly to us. Any non-certified funds will be held until proper clearance is made. COD orders are accepted as well as purchase orders from government agencies. UPS ground is our standard means of shipping unless otherwise specified. Shipping cost are available upon request. Tandy and Radio Shack are Trademarks of Tandy Corporation, TanPak, and TanTel are Trademarks of Hard Drive Specialist, a division of Compukit Corporation.

Printer News

Xerox Corporation's Xerox/Diablo D80 is a letter-quality daisy-wheel printer that prints at speeds up to 80 characters per second with noise measured as low as 58 decibels.

The D80 features Diablo's all-purpose interface (API), which allows easy hook-up to RS-232, Centronics, or IEEE 488 interfaces. A semiautomatic paper feed lets you switch between different paper sizes and weights.

The D80 uses Diablo's extended character set with 200 characters per print wheel. Multilingual print wheels are also available that allow the D80 to print in 33 languages. The printer is $2,195. An optional bidirectional tractor ($300) and a dual-bin, cut-sheet paper feeder ($903) are available. For more information, contact Xerox Corp., Xerox Square 006, Rochester, NY 14644, 716-423-5078.

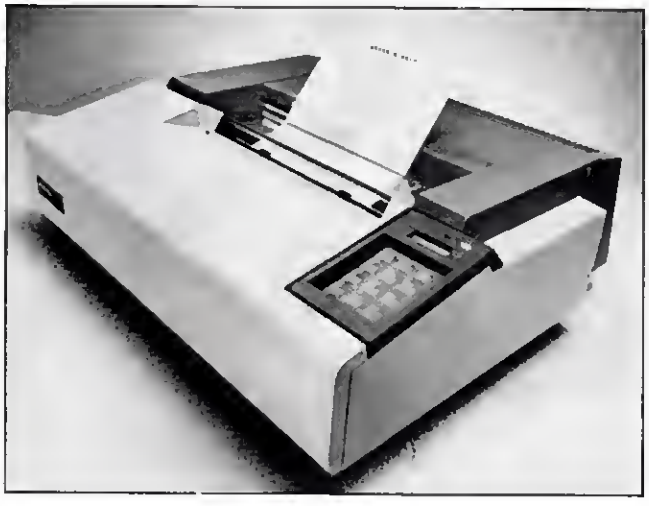
Circle 566 on Reader Service card.

Universal Basic

ZBasic 3.0 from Zedcor Inc. is faster than Turbo Pascal 3.0 and uses the same programming commands regardless of the computer you use. The IBM PC, PC XT, PC AT and compatibles; Apple IIc and IIe; Macintosh; TRS-80 Models I, III, 4; and CPM 80 2.2 and 3.0 computers all use the same commands with this language. ZBasic code works with all the leading microcomputers; if you write a program in ZBasic on an Apple, you can port it over to a Tandy computer and it will run the same way.

ZBasic includes device-independent graphics, 54-digit accuracy, a built-in interactive editor and compiler, a choice of alphanumeric labels or line numbers, and more at a base retail price of $89.95. For more information, contact Zedcor Inc., 3438 N. Country Club, Tucson, AZ 85716, 602-795-6800.

Circle 560 on Reader Service card.



The Xerox/Diablo D80 daisy-wheel printer is quiet and fast.

Board Talk

Fast80 ($59.95) from SOTA Computing Systems Ltd. is a bulletin board system for the Model 4/4P. It's written entirely in machine language so responses to user requests are almost instantaneous. The entire message/user/command base resides in memory. Fast80 needs 128K and works under TRSDOS 6.2 and DOSPLUS 4.

The bulletin board supports 445 different user IDs and handles up to 120 messages. Fast80 drives just about any direct-connect/auto-answer modem including Radio Shack's Modem II and Hayes and Microconnection units. For further information, contact SOTA Computing Systems Ltd., 213-1080 Broughton St., Vancouver, British Columbia, Canada, V6G 2A8, 604-688-5009.

Circle 554 on Reader Service card.

A, B and C

Jack Purdum's C *Self-Study Guide* helps you discover the C programming language at your own pace. Part I of the book provides questions of varying degrees of difficulty to guide beginners over the rough spots and to challenge more experienced C programmers. Part II has answers that include many complete programs for testing new functions and for illustrating tips, traps, techniques, and shortcuts.

The book is approximately 250 pages and costs $16.95. For more information, contact Que Corporation, 7999 Knue Road, Suite 202, Indianapolis, IN 46250, 800-428-5331.

Circle 562 on Reader Service card.

Remote Control

Dynatech's Turn-On, a compact, intelligent power controller, automatically activates unattended computer systems 24 hours a day, permitting file transfers, electronic mail delivery, and more.

Turn-On activates and powers up a computer system when it senses an incoming phone call, leaves the system on for the duration of the call, records the time of the call and all activity, and powers the system down when the call is completed. Turn-On comes with full communications software capability and built-in ac power and phone line protection to guard against power surges and potentially damaging voltage spikes.

Turn-On sells for $295. For more information, contact Dynatech Computer Power Inc., 4744 Scotts Valley Drive, Scotts Valley, CA 95066, 800-638-9098.

Circle 555 on Reader Service card.

Language Tutor

Learn Spanish The Easy Way ($69.95 plus $3 shipping) is part of a new foreign-language software series from International Computer Products that runs on the Models III and 4, some CP/M-80 based systems, and MS-DOS computers.

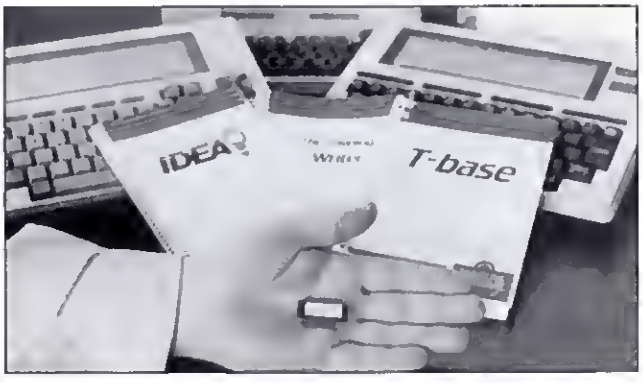
You type in a simple sentence in English. The program uses artificial intelligence techniques to interpret the sentence and display or print a Spanish translation. A large vocabulary allows phrase compositions with over 80,000 possible word combinations. The program also includes conjugation of regular and irregular verbs, and exercises in noun declension. A German version is also available. You can also get an optional voice synthesizer for the Models III and 4.

For more information, contact International Computer Products, 346 N. Western Ave., Los Angeles, CA 90004, 213-462-8381.

Circle 565 on Reader Service card.

Stocking Stuffer

The Floppy Disk Story is a 32-page booklet from Fuji Photo Film that introduces computer users of all ages to the basic building block of the floppy disk. The booklet teaches you about a floppy disk's construction and operation, as well as its proper care. While appealing to children, *The Floppy Disk Story* is also perfect for the not-so-young who are new to computing.



Three programs for the Models 100 and 200 come on one ROM chip.

You can get the booklet free by writing to Fuji Film Promotion Dept., P.O. Box 9870, Wethersfield, CT 06109. Limit requests to one per household.

Circle 556 on Reader Service card.

Three in One

Traveling Software has put three of its most popular Model 100/200 programs on a single ROM chip. The Ultimate ROM includes Idea!, an outline processor; T-Base, a data base management system; and T-Writer, a text formatter.

Because these three programs reside in ROM, they use almost no RAM, allowing larger data files than ever before. Ultimate ROM sells for $229.85 and includes the Traveling Memory Manager and an audio cassette overview of the programs.

The Ultimate ROM is available from Radio Shack stores. For more information, contact Traveling Software Inc., 11050 Fifth Ave. N.E., Seattle, WA 98125, 206-367-8090.

Circle 552 on Reader Service card.

Executive Covers

Regatta computer covers from Cover Craft Corp. are made of cotton/polyester blend material specially treated to resist stains, moisture, and wrinkles. They are also machine washable and nonstatic.

The covers come with a built-in disk storage pocket and pencil holder, and a storage wallet for the cover when not in use. Sizes are available for all popular microcomputers and related equipment. Colors are sand with navy trim and navy with sand trim. Prices range from $21.95 to $34.95.

For further information, contact Cover Craft Corp.,

Dress your computer in Regatta wear.

With TAS-41, four users can share one computer.

540 N. Commercial St., Manchester, NH 03101, 800-644-3555.

Circle 561 on Reader Service card.

Hook Them Up

Western Telematic Inc. offers a terminal-activated switch that lets up to four users share one computer port. The TAS-41 ($395) makes switching automatic. Each user connects and disconnects by entering simple commands from the terminal's keyboard. To prevent port tie-up in the event a user forgets to log-off, a timeout feature disconnects a port if no data activity takes place within a user-selectable time period.

For additional information, contact Western Telematic Inc., 2435 Anne St., Santa Ana, CA 92704, 800-854-7226.

Circle 559 on Reader Service card.

Power Play

American Power Conversion Corp. introduces the 450AT uninterruptible power source. When protecting a typical enhanced personal computer with a hard disk and monitor, the 450AT provides 15 minutes of operating time during an extended outage. For added protection, an electronic overcurrent protection and a master power switch control come standard.

The 450AT incorporates an internal, sealed, maintenance-free battery and a precision

The 450AT is specifically designed to power your desktop computer.

battery charger to insure dependable power at a moment's notice. It also offers built-in surge protection and combined EMI/RFI filtering.

The price is $799. For more details, contact American Power Conversion Corp., 89 Cambridge St., Burlington, MA 01803, 800-343-2507.

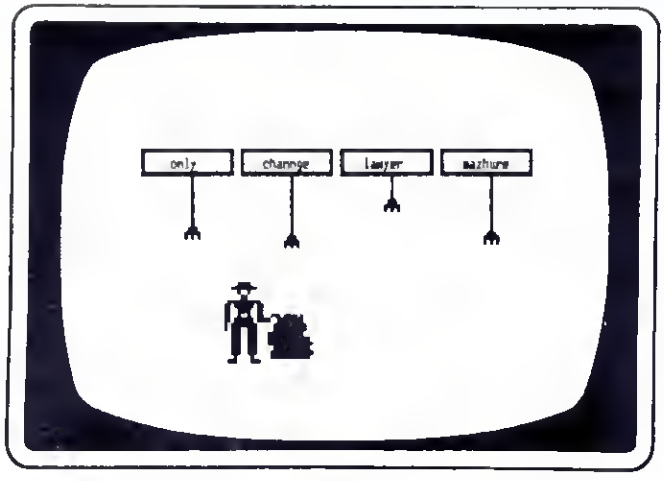
Circle 567 on Reader Service card.

Spider Terror

Gamco Industries' Spider Hunt Spelling game ($39.95) for the Models III and 4 teaches spelling and includes student and program management systems as well.

The student becomes a spider hunter whose goal is to catch as many spiders as possible in a set amount of time (one to 10 minutes). You can also set a level of difficulty (1–5).

Four words appear at the top of the screen, each with a spider descending from it. Some of the words are spelled correctly and some are misspelled. Students move the spider hunter from word to word and identify each word as correctly spelled or misspelled. If they answer correctly the spider falls into a sack.



Spider Hunt Spelling teaches spelling the fun way.

A bank of 400 words (200 spelled correctly and 200 misspelled) permanently resides on the disk. Teachers may also enter up to five lists of 80 words each.

For more details, contact Gamco Industries Inc., Box 1911, Big Spring, TX 79721, 800-351-1404.

Circle 558 on Reader Service card.

Facts and Logic

Logical Lynx from Krell Software Corp. teaches the art of scientific reasoning by showing you how scientists apply logic to actual bodies of knowledge in the natural and social sciences and humanities. It also teaches you basic scientific facts and how you can understand, organize, and fit these facts into meaningful patterns.

Twenty data bases provide basic information in critical areas of knowledge, as well as techniques for linking that information in new ways. You create, research, and organize new data bases on subjects of your choice.

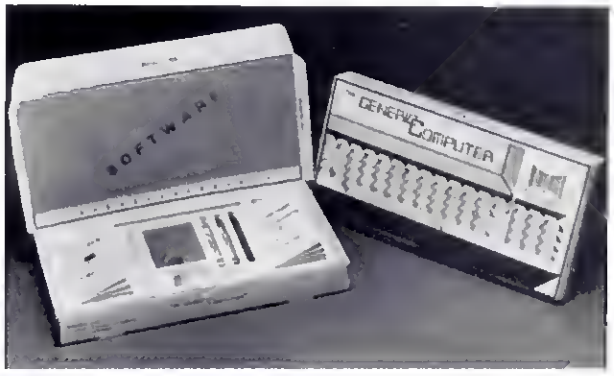
Logical Lynx comes in three levels of difficulty priced at $49.95, $69.95, and $89.95. One free data base comes with each system's master.

Additional data bases cost $19.95 each. Sample data bases include sports of the world, great writers, and chemical elements.

For more information, contact Krell Software Corp., 1320 Stony Brook Road, Stony Brook, NY 11790, 800-245-7355.

Circle 563 on Reader Service card.

DIFFERENT TRACK



The Generic Computer comes with 16K and color graphics.

New Generation Computer

The Generic Computer from Intec Inc. weighs less than 10 ounces and comes with color graphics, a letter-quality printer, its own special software, 16K, and the most dependable chip in the industry.

The computer is designed and written in good taste and sells for $5.95. Add 50 cents for shipping and handling. Contact Intec Inc. (676 Polk St., Eugene, OR 97402, 503-343-1464) for more information.

Circle 564 on Reader Service card.

New Products Index

Reader Service Number	Company	Page
567	American Power Conversion	148
561	Cover Craft Corp.	147
555	Dynatech Computer Power Inc.	146
556	Fuji Film Promotion Dept.	146
558	Gamco Industries Inc.	149
564	Intec Inc.	150
565	International Computer Products	146
563	Krell Software Corp.	149
562	Que Corp.	146
554	SOTA Computing Systems Limited	146
552	Traveling Software Inc.	147
559	Western Telematic Inc.	148
566	Xerox Corp.	146
560	Zedcor Inc.	146

New Products listings are based on information supplied in manufacturers' press releases. 80 Micro has not tested or reviewed these products and cannot guarantee any claims.

THE CLASSIFIED 80micro

SOFTWARE

MODEL 4/4P Ready-to-run Programs (Sales Files, Stats, Forecasts). Order "SALESPRO" $55. FORMTEC, PO Box D, Three Lakes, WI 54562

Master File – DBMS. Create hundreds of custom filing systems. Mail lists, inventory lists, customer records and hundreds of others. Print labels, lists, indexes, directories, cards and reports in any field order, sorted or unsorted. Ultra fast search routine. Choose any field to sort (ascending, descending, alphabetical, numeric), calculate or total. No programming required. Users say: "Simple yet powerful", "Easy to work with", "Superior to Profile", "Best information system I've seen". Only $59.00. Model-I, III, IV/IVP. MC – VISA
Ultimate Software, P.O. Box 1291, Hayden Lake, ID (208) 772-7634

PAYROLL MODEL 4/4P 128 Employees, 9 Earninga, 9 Deductions, Prints Paychecks, Check Register, Journals, W2's, California DE3, Employee list, etc. $75.00. Calif. orders add 6%. R8H Software 2950 Mission, #14 Solvang, CA 93463.

CANADIAN LOTTERIES Addresses and Winning BASIC Program for Models I/III/IV/CO-CO/1000 $5.00
SES MARKETING, POB 23518, Orlando, FL 32867

ANTI-ANTIDISESTABLISHMENTARIANISM. Cheap ads for YOUR New Used Software Hardware. Trash AD, Box 2804, Napa 94558.

SOFTWARE MODEL 3 & 4. HUNTLEY, 286 EUREKA STREET, S.F. CA 94114-2437.

BRAND NAME SOFTWARE and accessories for TRS-80's & PC's all discounted. Midwest Supply, Box 364-C, Clinton, Iowa 52732.

MEGASORT: Sorts megabyte MS-DOS text files. Outputs standard, limit, tag or keytag files. Multiple keys. Each key A/D. $125 Demo $10 Specify 1000/1200/2000. JW Luhrman 2nd floor 247 W. 38th Street NY, NY 10018 (212) 382-1722

Print graphics on the fly and more! Lightening 1.3 boot. Others. Send SASE for information. 11575 Sunshine, Studio City, California 91604-3835.

ISAM under BASIC. Multikey indexed files with your BASIC programs. Relocatable machine code runtime. Fast! Fully interactive utilities included. Model I, III, 4 LDOS. $140 or SASE for info. Terasoft, 34 Greenfield, Berea, OH 44017

HARDWARE

MODEL I/III HI-SPEED TAPE Interface $49.95, 14X faster. HHCI, 725 Idlewild, Bel Air, MD, 21014. 301-838-7692.

MODEMS, PRINTERS, TERMINALS Distributor pricing to end users and dealers—FREE SHIPPING CALL 1-800-833-2600 for catalog

---

64K RAM installed in Keyboard: TRS-80 M1-LII. Superior performance with or without EI. $59.95 with 90-day warranty. ICE; 100 Mill Street; Drakes Branch, VA 23937; (800) 424-3311.

DISKS & SUPPLIES

Diaks, Program Packaging Low prices on BASF and other disks. Binders, Slips like Tandy 1000, 2000. Much more! Free Catalog. Anthropomorphic, 376 East Saint Charles, Lombard, IL 60148 (312) 629-5160, 1-800-DEAL-NOW

SERVICES

NationServ Online Information Network. 818-847-2291. Jobservice, Downloads, Play Games, Shopping Mall, Classifieds, Massage Center, Etc.

Get the attention you deserve. *80 Micro* is now offering classified ads at a special introductory price. Reach over 100,000 readers with news of your product or service. Classified ads on the pages of *80 Micro* get results. Write to *80 Micro*, 80 Pine St., Peterborough, NH 03458 attn: Classified manager for information and deadlines.

Circle **849** on Reader Service Card

Get the attention you deserve. For only $5 per word, your ad will be seen by over 100,000 dedicated TRS-80/Tandy users. Ads must be received by the 20th of the month 3 months prior to publication date. **Send yours today.**

Please Print Carefully. We'll Do Our Best, But We Can't Be Responsible For Illegible Writing.

Name _____

Address _____

City _____ State _____ Zip _____

Telephone _____

CATEGORY _____

_____ _____ _____ (3 words)
_____ _____ _____ (6)
_____ _____ _____ (9)
_____ _____ _____ (12)
_____ _____ _____ (15)
_____ _____ _____ (18)
_____ _____ _____ (21)
_____ _____ _____ (24)
_____ _____ _____ (27)

Number of Words _____

Total × $5/issue = _____

For _____ issue

Make Checks Payable to 80 Micro
NO AD ACCEPTED WITHOUT PAYMENT
80 Pine St., Peterborough, NH 03458

Little Treasures

Welcome to Fine Lines, *80 Micro*'s new back-page contest. The purpose of this column is to give you a chance to flex your programming muscles, learn some techniques from other readers, and (most importantly) win a prize or two. Each month, we'll give you a problem that needs solving, along with the winning solutions to a previous month's problem. If we publish your solution, you'll receive (at the very least) an "I Break for *80 Micro*" bumper sticker. If we think you've demonstrated particular brilliance and creativity, we'll send you an *80 Micro* T-shirt (don't count on it, though; we're stingy with the T-shirts).

If you can't seem to solve the problems, don't despair; we'll hand out prizes for any contest ideas you submit that we use. Since this is the first installment of Fine Lines, here's a run-down of the rules:

1. Owners of all TRS-80 and Tandy systems with the exception of the Pocket Computers are eligible. We'll consider degree of difficulty when comparing solutions created on different machines.
2. The deadline will always be the 21st of the issue month. Thus, this month's deadline is Dec. 21. We realize that this doesn't give everyone the same amount of time to come up with their entries (we apologize to our overseas readers especially), but postponing the deadline any longer would add another month to our publishing the answers.
3. Speaking of the answers, they'll appear three issues from the issue in which the problem appears. Thus, this month's winners will make their appearance in the March 1986 issue.
4. Employees of CW Communications are not eligible.
5. We will not, unfortunately, be able to return entries.
6. Specify your T-shirt size. Bumper size not required.

Contest No. 1

Okay, enough of that. Let's get down to the first contest. Your task is to write a word processor in two lines of Basic. Simple enough, eh?

We'll judge entries on the basis of creativity, number of features, and programming elegance.

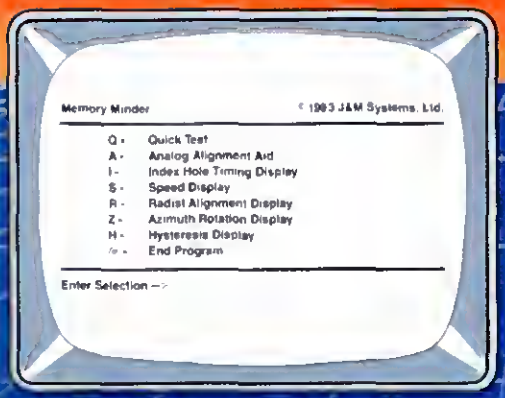
Just to get you going, we whipped up the clunky little Model 4 program you see in the Program Listing. We're confident that you can come up with something better. Give it a try; you might win yourself a T-shirt.■

Program Listing. 80 Micro's wimpy little two-line word processor.

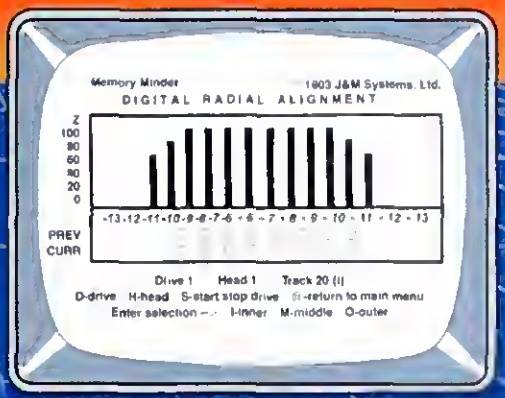
```
11# WHILE X$<>CHR$(0):X=0:X$=INKEY$:IF X$="" THEN GOTO 11# ELSE IF X$=CHR$(9) TH
EN X$=CHR$(25) ELSE IF X$=CHR$(10) THEN X$=CHR$(26) ELSE IF X$=CHR$(11) THEN X$=
CHR$(27)
2## PRINT X$;:WEND
```

MEMORY MINDER™

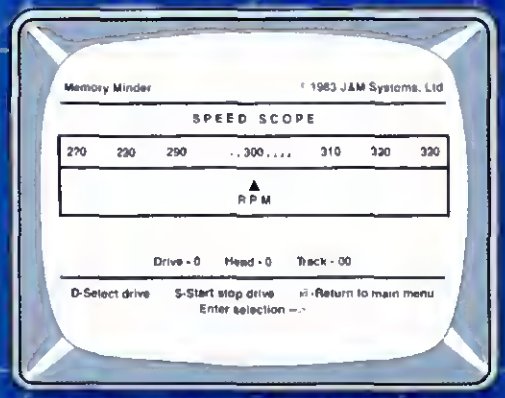
. . . A UNIQUE APPROACH TO DISK RELIABILITY!



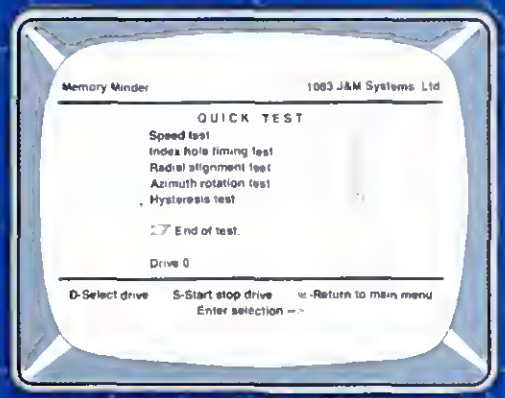
Select any one of seven tests to perform preventive maintenance or to isolate problems. Simple, single-letter commands make MM easy to use! Use MM to align the head, adjust the index hole detector, or adjust the speed.



Use the MM Radial Alignment Test to check the head alignment of your drives. No need for an oscilloscope or other expensive test equipment!



Check the motor speed of your drives. Or, you can even use the Speed Test to adjust the drive speed. No need for any test equipment!



Use the Quick Test to quickly and automatically test five of the most important performance parameters of your drive. Monitor your drives for long term drift. Isolate problems quickly and automatically!

PROTECT YOUR DATA.

Now you can make sure your data is being recorded properly by the use of the revolutionary ***Memory Minder.***

The *Memory Minder* from J & M Systems, tests your disk's performance and calibration without any additional equipment! It measures your disk's performance and displays it on your screen.

This is the most comprehensive disk diagnostic program available for your TRS-80 microcomputer. You can even adjust drive alignment while watching the display!

Spot problems ***before*** they endanger your data! If you own a disk drive, you ***need*** the ***Memory Minder!***

	Price
TRS-80 Model III/4	
1 - 51 tpi Single Side (Standard)	$79
2 - 48 tpi Double Side	$89
3 - 96 tpi Double Side	$129
Includes 5¼ tpi & 8" tpi program diskettes	
TRS-80 Model I	
1 - 48 tpi Single Side Single Density	$89
TRS-80 Color Computer and TDP-100	
1 - 48 tpi Single Side (Standard)	$79
2 - 48 tpi Double Side	$99

MM also available for other models

J & M SYSTEMS IS THE DRIVING FORCE!

J & M SYSTEMS, LTD.
15100-A CENTRAL SE
ALBUQUERQUE, NEW MEXICO 87123
505/292-4182

Circle 41 on Reader Service card.

Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • OSBORNE™ • CP/M™ • KAYPRO™

FREE--TRS-80 Mod I,III & 4 programs supplied on DosPlus (minimum system). Complete DosPlus also available

The VERSABUSINESS™ Series

Each VERSABUSINESS module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated business system.

VERSARECEIVABLES™ $99.95

VERSARECEIVABLES™ is a complete menu-driven accounts receivable, invoicing, and monthly statement generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VERSARECEIVABLES™ prints all necessary statements, invoices, and summary reports and can be linked with VERSALEDGER II™ and VERSAINVENTORY™.

VERSAPAYABLES™ $99.95

VERSAPAYABLES™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VERSAPAYABLES™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VERSAPAYABLES™, you can even let your computer automatically select which vouchers are to be paid.

VERSAPAYROLL™ $99.95

VERSAPAYROLL™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VERSALEDGER II™ system.

VERSAINVENTORY™ $99.95

VERSAINVENTORY™ is a complete inventory control system that gives you instant access to data on any item. VERSAINVENTORY™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VERSARECEIVABLES™ system. VERSAINVENTORY™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

VERSALEDGER II™ $149.95

VERSALEDGER II™ is a complete accounting system that grows as your business grows. VERSALEDGER II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software.**

- VERSALEDGER II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system).
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VERSALEDGER II™ comes with a professionally-written 160 page manual designed for first-time users. The VERSALEDGER II™ manual will help you become quickly familiar with VERSALEDGER II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

SATISFACTION GUARANTEED!

Every VERSABUSINESS™ module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any VERSABUSINESS™ module, you may return it within 30 days for a refund. Manuals for any VERSABUSINESS™ module may be purchased for $25 each, credited toward a later purchase of that module.
All CP/M based Computers must be equipped with Microsoft BASIC (MBASIC or BASIC 80)

To Order:
Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

- add $3 for shipping in UPS areas
- add $4 for C.O.D. or non-UPS areas
- add $5 to CANADA or MEXICO
- add proper postage elsewhere

DEALER INQUIRIES WELCOME
All prices and specifications subject to change. Delivery subject to availability.

COMPUTRONICS INC

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977

TRS-80 trademark Tandy Corp · APPLE trademark Apple Corp · IBM PC trademark IBM Corp · OSBORNE trademark Osborne Corp. · XEROX trademark Xerox Corp · KAYPRO trademark Non-Linear Systems, Inc · TELEVIDEO trademark Televideo Systems, Inc · SANYO trademark Sanyo Corp · NEC trademark NEC Corp · DEC trademark Digital Equipment Corp · ZENITH trademark Zenith Corp · TI PROFESSIONAL COMPUTER trademark Texas Instruments, Inc · SUPERBRAIN trademark Intertec Corp · CP/M trademark Digital Research · EPSTON trademark Epson Corp